

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-305923

(43)Date of publication of application : 02.11.2000

(51)Int.Cl. G06F 17/16
G06F 15/16

(21)Application number : 11-113386

(71)Applicant : FUJI XEROX CO LTD
TAISHO PHARMACEUT CO LTD

(22)Date of filing : 21.04.1999

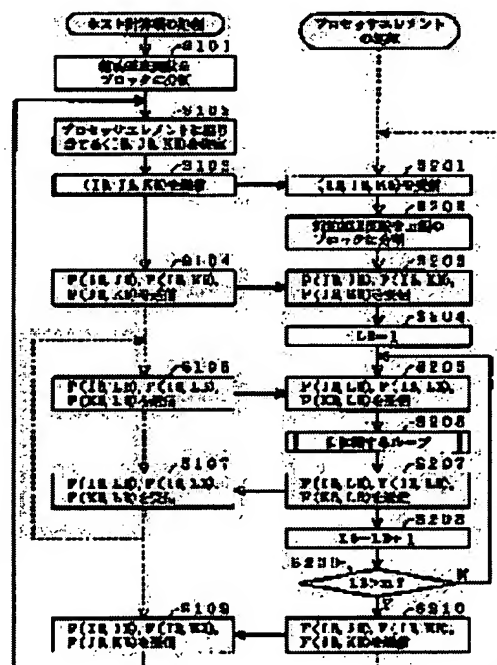
(72)Inventor : YAMADA SOU
INAHATA SHINJIROU
MIYAGAWA NOBUAKI
TAKASHIMA HAJIME
KITAMURA KAZUYASU

(54) METHOD FOR PARALLEL CALCULATION OF MATRIX ELEMENT AND METHOD FOR CALCULATION OF MOLECULAR ORBIT

(57)Abstract:

PROBLEM TO BE SOLVED: To provide a parallel processing method which can perform efficient matrix element calculation irrelevantly to the communication performance between a host computer and a processor element by sending at least some of matrix elements corresponding to indexes included in a block to a host computer.

SOLUTION: The transmission of density matrix element blocks $P(IB, LB)$, $P(JB, LB)$, and $P(KB, LB)$ having $I_c \times I_p$ elements is requested and they are received (S205). After a loop is completed and the calculation of one LB ends, the processor element sends fox matrix element blocks $F(IB, LB)$, $F(JB, LB)$, and $F(KB, LB)$ having $I_c \times I_p$ elements back to a host computer (S207). When it is judged (S209) that those processes are completed for all LB , the processor element sends the fox matrix element blocks having $I_c \times I_c$ elements back to the host computer (S210).



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision
of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開2000-305923

(P 2 0 0 0 - 3 0 5 9 2 3 A)

(43) 公開日 平成12年11月2日(2000.11.2)

| | | | |
|----------------------------|------|-------------|-------------|
| (51) Int. Cl. ⁷ | 識別記号 | F I | テーマコード (参考) |
| G06F 17/16 | | G06F 15/347 | M 5B045 |
| 15/16 | 610 | 15/16 | G 5B056 |

審査請求 未請求 請求項の数10 O L (全27頁)

(21) 出願番号 特願平11-113386

(22) 出願日 平成11年4月21日(1999.4.21)

(71) 出願人 000005496

富士ゼロックス株式会社
東京都港区赤坂二丁目17番22号

(71) 出願人 000002819

大正製薬株式会社
東京都豊島区高田3丁目24番1号

(72) 発明者 山田 想

神奈川県足柄上郡中井町境430 グリーン
テクなかい富士ゼロックス株式会社内

(74) 代理人 100091546

弁理士 佐藤 正美

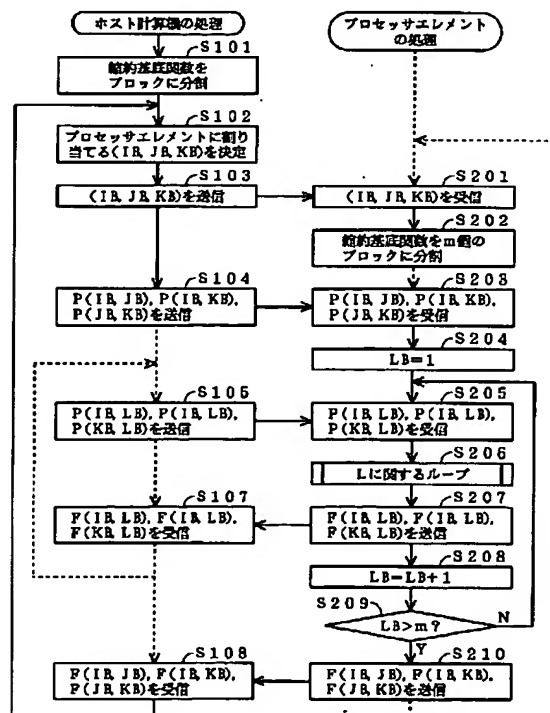
最終頁に続く

(54) 【発明の名称】 行列要素の並列計算方法および分子軌道計算方法

(57) 【要約】

【課題】 安価な通信手段と、小容量のメモリを有する多数のプロセッサエレメントとを用いた並列計算によっても、ホスト計算機とプロセッサエレメントとの間の通信性能に律速されることなく、効率的に分子軌道計算を行える。

【解決手段】 ホスト計算機で、縮約基底関数のインデックスを複数のブロックに分割し、そのブロックを指示する3つの整数インデックス I B, J B, K B に関する所定の3重ループを形成するとともに、それらインデックス I B, J B, K B で指定されたものを単位として1つのジョブを形成する。そして、そのジョブをプロセッサエレメントに割り当て、必要な密度行列要素を、そのプロセッサエレメントに送信する。プロセッサエレメントは、縮約シェルを複数のブロックに分割し、そのブロックを指示する整数インデックス L B に関するループ制御を行うとともに、インデックス L B が切り替わる毎に、2電子積分の関数値の一部およびフォック行列要素の一部を計算し、フォック行列要素の一部をホスト計算機に送信する。



【特許請求の範囲】

【請求項1】 同じ1からNの範囲にある4つの整数インデックスI, J, K, Lを用いて表わされ、 $G(I, J, K, L) = G(I, J, L, K) = G(J, I, K, L) = G(J, I, L, K) = G(K, L, I, J) = G(K, L, J, I) = G(L, K, I, J) = G(L, K, J, I)$ なる関係を満たす関数Gの関数値 $G(I, J, K, L)$ と；2つの前記整数インデックスK, Lを用いて表わされ、 $P(K, L) = P(L, K)$ なる関係を満たす行列Pの要素 $P(K, L)$ と；係数 A_1 と；の積 $A_1 \cdot P(K, L) \cdot G(I, J, K, L)$ についての前記範囲の全ての前記Kおよび前記Lに関する総和 $F_1(I, J)$ と、
 前記関数値 $G(I, L, K, J)$ と；前記行列要素 $P(K, L)$ と；係数 A_2 と；の積 $A_2 \cdot P(K, L) \cdot G(I, L, K, J)$ に関する前記範囲の全ての前記Kおよび前記Lにおける総和 $F_2(I, J)$ との和 $F(I, J) = F_1(I, J) + F_2(I, J)$ を要素とする行列Fの全要素を、ホスト計算機および複数のプロセッサエレメントにより構成される並列計算装置を用いて計算する行列要素の並列計算方法において、
 前記ホスト計算機で前記1からNの範囲にある前記インデックスを複数のブロックに分割して、第1のブロック群を形成し、
 前記ホスト計算機で前記第1のブロック群のブロックを指示する3つの整数インデックスIB, JB, KBに関する所定の3重ループを形成し、
 前記ホスト計算機で前記ブロックを指示する前記3つの整数インデックスIB, JB, KBで指定されたものを単位として1つのジョブを形成し、
 前記ホスト計算機で前記1つのジョブを、前記並列計算装置内の一つの前記プロセッサエレメントに対して割り当て、
 前記ホスト計算機は、
 前記ジョブの割り当てを行う際に、
 前記ブロックIBに含まれる前記インデックスI、および前記ブロックJBに含まれる前記インデックスJ、および前記ブロックKBに含まれる前記インデックスK、に対応する行列要素 $P(I, J)$ および $P(I, K)$ および $P(J, K)$ の少なくとも一部を前記プロセッサエレメントに対して送信し、
 前記プロセッサエレメントは、
 前記1からNの範囲にある前記整数インデックスを、前記第1のブロック群と同じまたは異なる複数のブロックに分割して、第2のブロック群を形成して、その前記第2のブロック群のブロックを指示するインデックスLBに関するループ制御を行うとともに、
 前記インデックスLBが切り替わる毎に、
 前記ブロックIBに含まれる前記インデックスI、および前記ブロックJBに含まれる前記インデックスJ、お

よび前記ブロックKBに含まれる前記インデックスK、および前記ブロックLBに含まれる前記インデックスL、にそれぞれ対応する行列要素 $P(I, L)$ および $P(J, L)$ および $P(K, L)$ の少なくとも一部を前記ホスト計算機から受信し、
 前記ブロックIBに含まれる前記インデックスI、および前記ブロックJBに含まれる前記インデックスJ、および前記ブロックKBに含まれる前記インデックスK、および前記ブロックLBに含まれる前記インデックスL、にそれぞれ対応する関数値 $G(I, J, K, L)$ の少なくとも一部を計算し、
 前記計算された関数値 $G(I, J, K, L)$ を用いて、
 前記ブロックIBに含まれる前記インデックスI、および前記ブロックJBに含まれる前記インデックスJ、および前記ブロックKBに含まれる前記インデックスK、および前記ブロックLBに含まれる前記インデックスL、に対応する行列要素 $F(I, J)$ および $F(I, K)$ および $F(J, K)$ および $F(I, L)$ および $F(J, L)$ および $F(K, L)$ の少なくとも一部を所定の計算式に従って計算し、
 前記ブロックIBに含まれる前記インデックスI、および前記ブロックJBに含まれる前記インデックスJ、および前記ブロックKBに含まれる前記インデックスK、および前記ブロックLBに含まれる前記インデックスL、に対応する行列要素 $F(I, L)$ および $F(J, L)$ および $F(K, L)$ の少なくとも一部を前記ホスト計算機へ送信し、
 さらに、前記プロセッサエレメントは、
 前記ジョブの終了時に、
 前記ブロックIBに含まれる前記インデックスI、および前記ブロックJBに含まれる前記インデックスJ、および前記ブロックKBに含まれる前記インデックスK、にそれぞれ対応する行列要素 $F(I, J)$ および $F(I, K)$ および $F(J, K)$ の少なくとも一部を前記ホスト計算機へ送信することを特徴とする行列要素の並列計算方法。
 【請求項2】 同じ1からNの範囲にある4つの整数インデックスI, J, K, Lを用いて表わされ、 $G(I, J, K, L) = G(I, J, L, K) = G(J, I, K, L) = G(J, I, L, K) = G(K, L, I, J) = G(K, L, J, I) = G(L, K, I, J) = G(L, K, J, I)$ なる関係を満たす関数Gの関数値 $G(I, J, K, L)$ と；2つの前記整数インデックスK, Lを用いて表わされ、 $P(K, L) = P(L, K)$ なる関係を満たす行列Pの要素 $P(K, L)$ と；係数 A_1 と；の積 $A_1 \cdot P(K, L) \cdot G(I, J, K, L)$ についての前記範囲の全ての前記Kおよび前記Lに関する総和 $F_1(I, J)$ と、
 前記関数値 $G(I, L, K, J)$ と；前記行列要素 $P(K, L)$ と；係数 A_2 と；の積 $A_2 \cdot P(K, L) \cdot$

G (I , L , K , J) に関する前記範囲の全ての前記 K および前記 L における総和 $F2(I, J)$ との和 $F(I, J) = F1(I, J) + F2(I, J)$ を要素とする行列 F の全要素を、

ホスト計算機および複数のプロセッサエレメントにより構成される並列計算装置を用いて計算する行列要素の並列計算方法において、

前記ホスト計算機で前記 1 から N の範囲にある前記インデックスを複数のブロックに分割して、第 1 のブロック群を形成し、

前記ホスト計算機で前記第 1 のブロック群のブロックを指示する 3 つの整数インデックス I B , J B , K B に関する所定の 3 重ループを形成し、

前記ホスト計算機で前記ブロックを指示する前記 3 つの整数インデックス I B , J B , K B が固定されたものを単位として 1 つのジョブを形成し、

前記ホスト計算機で前記 1 つのジョブを前記並列計算装置内の一つのプロセッサエレメントに対して割り当て、前記ホスト計算機は、

前記ジョブの割り当てを行う際に、

前記ブロック I B に含まれる前記インデックス I , および前記ブロック J B に含まれる前記インデックス J , および前記ブロック K B に含まれる前記インデックス K , に対応する行列要素 $P(I, J)$ および $P(I, K)$ および $P(J, K)$ の少なくとも一部を前記プロセッサエレメントに対して送信し、

前記プロセッサエレメントは、

前記 1 から N の範囲にある前記インデックスを、前記第 1 のブロック群と同じまたは異なる複数のブロック群に分割して、第 2 のブロック群を形成して、その前記第 2 のブロック群のブロックを指示するインデックス L B に関するループ制御を行うとともに、

前記インデックス L B が切り替わる毎に、

前記ブロック I B に含まれる前記インデックス I , および前記ブロック J B に含まれる前記インデックス J , および前記ブロック K B に含まれる前記インデックス K , および前記ブロック L B に含まれる前記インデックス L , に対応する行列要素 $P(I, L)$ および $P(J, L)$ および $P(K, L)$ の少なくとも一部を、前記ホスト計算機から受信し、

前記ブロック I B に含まれる前記インデックス I , および前記ブロック J B に含まれる前記インデックス J , および前記ブロック K B に含まれる前記インデックス K , および前記ブロック L B に含まれる前記インデックス L , に対応する関数値 $G(I, J, K, L)$ および $G(I, K, J, L)$ および $G(I, L, J, K)$ の少なくとも一部を計算し、

前記計算された関数値を用いて、前記ブロック I B に含まれる前記インデックス I , および前記ブロック J B に含まれる前記インデックス J , および前記ブロック K B に含まれる前記インデックス K , および前記ブロック L

B に含まれる前記インデックス L , に対応する行列要素 $F(I, J)$ および $F(I, K)$ および $F(J, K)$ および $F(I, L)$ および $F(J, L)$ および $F(K, L)$ の少なくとも一部を所定の計算式に従って計算し、前記ブロック I B に含まれる前記インデックス I , および前記ブロック J B に含まれる前記インデックス J , および前記ブロック K B に含まれる前記インデックス K , および前記ブロック L B に含まれる前記インデックス L , に対応する行列要素 $F(I, L)$ および $F(J, L)$ および $F(K, L)$ の少なくとも一部を、前記ホスト計算機へ送信し、

さらに、前記プロセッサエレメントは、

前記ジョブの終了時に、

前記ブロック I B に含まれる前記インデックス I , および前記ブロック J B に含まれる前記インデックス J , および前記ブロック K B に含まれる前記インデックス K , に対応する行列要素 $F(I, J)$ および $F(I, K)$ および $F(J, K)$ の少なくとも一部を、前記ホスト計算機へ送信することを特徴とする行列要素の並列計算方法。

【請求項 3】請求項 1 に記載の行列要素の並列計算方法において、

前記プロセッサエレメントの処理における前記インデックス L B の切り替わり時に、前記ホスト計算機から前記プロセッサエレメントへ送信されるのは、

前記ブロック K B に含まれる前記インデックス K および前記ブロック L B に含まれる前記インデックス L に対応する行列要素 $P(K, L)$ のうち、前記関数値 $G(I, J, K, L)$ の計算が不必要であると判断されない前記インデックス K , L の組み合わせに対応する行列要素 $P(K, L)$ であることを特徴とする行列要素の並列処理方法。

【請求項 4】請求項 1 に記載の行列要素の並列計算方法において、

前記プロセッサエレメントの処理における前記インデックス L B の切り替わり時に、前記ホスト計算機から前記プロセッサエレメントへ送信されるのは、

前記ブロック I B に含まれる前記インデックス I および前記ブロック J B に含まれる前記インデックス J および前記ブロック K B に含まれる前記インデックス K および前記ブロック L B に含まれる前記インデックス L に対応する行列要素 $P(I, L)$ および $P(J, L)$ および $P(K, L)$ のうち、前記関数値 $G(I, J, K, L)$ の計算が不必要であると判断されない前記インデックス K , L の組み合わせに対応する、前記行列要素 $P(K, L)$ と、

前記ブロック I B に含まれる前記インデックス I および前記ブロック L B に含まれる前記インデックス L に対応する前記行列要素 $P(I, L)$ と、

前記ブロック J B に含まれる前記インデックス J および

前記ブロックLBに含まれる前記インデックスLに対応する前記行列要素 $P(J, L)$ であることを特徴とする行列要素の並列計算方法。

【請求項5】請求項2に記載の行列要素の並列計算方法において、

前記プロセッサエレメントの処理における前記インデックスLBの切り替わり時に、前記ホスト計算機から前記プロセッサエレメントへ送信されるのは、

前記ブロックIBに含まれる前記インデックスIおよび前記ブロックJBに含まれる前記インデックスJおよび前記ブロックKBに含まれる前記インデックスKおよび前記ブロックLBに含まれる前記インデックスLに対応する行列要素 $P(I, L)$ および $P(J, L)$ および $P(K, L)$ のうち、

前記関数値 $G(I, J, K, L)$ の計算が不必要であると判断されないか、あるいは前記ブロックJB内の全ての前記インデックスJとLとの組み合わせで前記関数値 $G(I, K, J, L)$ の計算が不必要であると判断されないか、あるいは前記ブロックIB内の全ての前記インデックスIとLとの組み合わせで前記関数値 $G(I, L, J, K)$ の計算が不必要であると判断されない前記行列要素 $P(K, L)$ と、

前記関数値 $G(I, L, J, K)$ の計算が不必要であると判断されないか、あるいは前記ブロックJB内の全ての前記インデックスJとLとの組み合わせで前記関数値 $G(I, K, J, L)$ の計算が不必要であると判断されないか、あるいは前記ブロックKB内の全ての前記インデックスKとLとの組み合わせで前記関数値 $G(I, J, K, L)$ の計算が不必要であると判断されない前記行列要素 $P(I, L)$ と、

前記関数値 $G(I, K, J, L)$ の計算が不必要であると判断されないか、あるいは前記ブロックIB内の全ての前記インデックスIとLとの組み合わせで前記関数値 $G(I, L, J, K)$ の計算が不必要であると判断されないか、あるいは前記ブロックKB内の全ての前記インデックスKとLとの組み合わせで前記関数値 $G(I, J, K, L)$ の計算が不必要であると判断されない前記行列要素 $P(J, L)$ であることを特徴とする行列要素の並列計算方法。

【請求項6】それぞれN個(Nは正の整数)の縮約シェルR, S, T, Uのそれぞれに含まれる原始シェルr, s, t, uのそれぞれの成分である原始基底関数i, j, k, lをインデックスとして用いて表わされる2電子積分関数gの関数値 $g(i, j, k, l)$ と;前記原始基底関数kをひとつの構成要素とする縮約基底関数Kおよび前記原始基底関数lをひとつの構成要素とする縮約基底関数Lをインデックスとして用いて表わされる密度行列Pの要素 $P(K, L)$ と;係数A1と;の積 $A1 \cdot P(K, L) \cdot g(i, j, k, l)$ の全ての縮約基底関数に関する総和 $f1(I, J)$ と、

前記2電子積分関数gの関数値 $g(i, k, j, l)$ と;前記密度行列Pの前記要素 $P(K, L)$ と;係数A2と;の積 $A2 \cdot P(K, L) \cdot g(i, k, j, l)$ の全ての縮約基底関数に関する総和 $f2(I, J)$ との和 $f(I, J) = f1(I, J) + f2(I, J)$ の、前記原始基底関数i, 前記原始基底関数jをそれぞれひとつの構成要素とする縮約基底関数I, 縮約基底関数Jに含まれる全ての原始基底関数に関する和で表わされるFock行列の全ての要素 $F(I, J)$ の計算を、ホスト計算機および複数のプロセッサエレメントにより構成される並列計算装置を用いて計算する分子軌道計算方法において、

前記ホスト計算機で前記1からNの範囲にある縮約基底関数の前記インデックスを複数のブロックに分割して、第1のブロック群を形成し、

前記ホスト計算機で前記第1のブロック群のブロックを指示する3つの整数インデックスIB, JB, KBに関する所定の3重ループを形成し、

前記ホスト計算機で前記ブロックを指示する前記3つの整数インデックスIB, JB, KBで指定されたものを単位として1つのジョブを形成し、

前記ホスト計算機で前記1つのジョブを、前記並列計算装置内の一つの前記プロセッサエレメントに対して割り当て、

前記ホスト計算機は、

前記ジョブの割り当てを行う際に、

前記ブロックIBに含まれる前記縮約基底関数I、および前記ブロックJBに含まれる前記縮約基底関数J、および前記ブロックKBに含まれる前記縮約基底関数K、に対応する密度行列要素 $P(I, J)$ および $P(I, K)$ および $P(J, K)$ の少なくとも一部を、前記プロセッサエレメントに対して送信し、

前記プロセッサエレメントは、

前記1からNの範囲にある縮約シェルを、前記第1のブロック群と同じまたは異なる複数のブロックに分割して、第2のブロック群を形成して、その前記第2のブロック群のブロックを指示する整数インデックスLBに関するループ制御を行うとともに、

前記インデックスLBが切り替わる毎に、

前記ブロックIBに含まれる前記縮約基底関数I、および前記ブロックJBに含まれる前記縮約基底関数J、および前記ブロックKBに含まれる前記縮約基底関数K、および前記ブロックLBに含まれる前記縮約基底関数L、に対応する密度行列要素 $P(I, L)$ および $P(J, L)$ および $P(K, L)$ の少なくとも一部を、前記ホスト計算機から受信し、

前記ブロックIBに含まれる前記縮約基底関数Iを構成する前記原始基底関数i、および前記ブロックJBに含まれる前記縮約基底関数Jを構成する前記原始基底関数j、および前記ブロックKBに含まれる前記縮約基底関

数 K を構成する前記原始基底関数 k 、および前記ブロック L に含まれる前記縮約基底関数 l を構成する前記原始基底関数 l に対応する2電子積分の関数値 $g(i, j, k, l)$ の少なくとも一部を計算し、その計算により求められた前記関数値を用いて、前記ブロック I に含まれる前記縮約基底関数 I 、および前記ブロック J に含まれる前記縮約基底関数 J 、および前記ブロック K に含まれる前記縮約基底関数 K 、および前記ブロック L に含まれる前記縮約基底関数 L 、に対応するフォック行列要素 $F(I, J)$ および $F(I, K)$ および $F(J, K)$ および $F(I, L)$ および $F(J, L)$ および $F(K, L)$ の少なくとも一部を所定の計算式に従って計算し、前記ブロック I に含まれる前記縮約基底関数 I 、および前記ブロック J に含まれる前記縮約基底関数 J 、および前記ブロック K に含まれる前記縮約基底関数 K 、および前記ブロック L に含まれる前記縮約基底関数 L 、に対応するフォック行列要素 $F(I, L)$ および $F(J, L)$ および $F(K, L)$ の少なくとも一部を前記ホスト計算機へ送信し、さらに、前記プロセッサエレメントは、前記ジョブの終了時に、前記ブロック I に含まれる前記縮約基底関数 I 、および前記ブロック J に含まれる前記縮約基底関数 J 、および前記ブロック K に含まれる前記縮約基底関数 K 、に対応するフォック行列要素 $F(I, J)$ および $F(I, K)$ および $F(J, K)$ の少なくとも一部を前記ホスト計算機へ送信することを特徴とする分子軌道計算方法。

【請求項7】それぞれ N 個(N は正の整数)の縮約シェル R, S, T, U のそれぞれに含まれる原始シェル r, s, t, u のそれぞれの成分である原始基底関数 i, j, k, l をインデックスとして用いて表わされる2電子積分関数 g の関数値 $g(i, j, k, l)$ と；前記原始基底関数 k をひとつの構成要素とする縮約基底関数 K および前記原始基底関数 l をひとつの構成要素とする縮約基底関数 L をインデックスとして用いて表わされる密度行列 P の要素 $P(K, L)$ と；係数 $A1$ との積 $A1 \cdot P(K, L) \cdot g(i, j, k, l)$ の全ての縮約基底関数に関する総和 $f1(I, J)$ と、前記2電子積分関数 g の関数値 $g(i, k, j, l)$ と；前記密度行列 P の前記要素 $P(K, L)$ と；係数 $A2$ と；の積 $A2 \cdot P(K, L) \cdot g(i, k, j, l)$ の全ての縮約基底関数に関する総和 $f2(I, J)$ との和 $f(I, J) = f1(I, J) + f2(I, J)$ の、前記原始基底関数 i 、前記原始基底関数 j をそれぞれひとつの構成要素とする縮約基底関数 I 、縮約基底関数 J に含まれる全ての原始基底関数に関する和で表わされるFock行列の全ての要素 $F(I, J)$ の計算を、ホスト計算機および複数のプロセッサエレメントにより構成

される並列計算装置を用いて計算する分子軌道計算方法において、前記ホスト計算機で前記1から N の範囲にある縮約基底関数の前記インデックスを複数のブロックに分割して、第1のブロック群を形成し、前記ホスト計算機で前記第1のブロック群のブロックを指示する3つの整数インデックス I, J, K に関する所定の3重ループを形成し、前記ホスト計算機で前記ブロックを指示する前記3つのインデックス I, J, K で指定されたものを単位として1つのジョブを形成し、前記ホスト計算機で前記1つのジョブを前記並列計算装置内の一つのプロセッサエレメントに対して割り当て、前記ホスト計算機は、前記ジョブの割り当てを行う際に、前記ブロック I に含まれる前記縮約基底関数 I 、および前記ブロック J に含まれる前記縮約基底関数 J 、および前記ブロック K に含まれる前記縮約基底関数 K 、に対応する密度行列要素 $P(I, J)$ および $P(I, K)$ および $P(J, K)$ の少なくとも一部を、前記プロセッサエレメントに対して送信し、前記プロセッサエレメントは、前記1から N の範囲にある縮約シェルを、前記第1のブロック群と同じまたは異なる複数のブロック群に分割して、第2のブロック群を形成して、その前記第2のブロック群のブロックを指示するインデックス L に関するループ制御を行うとともに、前記インデックス L が切り替わる毎に、前記ブロック I に含まれる前記縮約基底関数 I 、および前記ブロック J に含まれる前記縮約基底関数 J 、および前記ブロック K に含まれる前記縮約基底関数 K 、および前記ブロック L に含まれる前記縮約基底関数 L 、に対応する密度行列要素 $P(I, L)$ および $P(J, L)$ および $P(K, L)$ の少なくとも一部を、前記ホスト計算機から受信し、前記ブロック I に含まれる前記縮約基底関数 I を構成する前記原始基底関数 i 、および前記ブロック J に含まれる前記縮約基底関数 J を構成する前記原始基底関数 j 、および前記ブロック K に含まれる前記縮約基底関数 K を構成する前記原始基底関数 k 、および前記ブロック L に含まれる前記縮約基底関数 L を構成する前記原始基底関数 l 、に対応する2電子積分 $g(i, j, k, l)$ および $g(i, k, j, l)$ および $g(i, l, j, k)$ の少なくとも一部を計算し、この計算された前記関数値を用いて、前記ブロック I に含まれる前記縮約基底関数 I 、および前記ブロック J に含まれる前記縮約基底関数 J 、および前記ブロック K に含まれる前記縮約基底関数 K 、および前記ブロック L に含まれる前記縮約基底関数 L 、に対応するフォック行列要素 $F(I, J)$ および $F(I, K)$ および F

(J , K) および F (I , L) および F (J , L) および F (K , L) の少なくとも一部を所定の計算式に従って計算し、

前記ブロック I B に含まれる前記縮約基底関数 I 、および前記ブロック J B に含まれる前記縮約基底関数 J 、および前記ブロック K B に含まれる前記縮約基底関数 K 、および前記ブロック L B に含まれる前記縮約基底関数 L 、に対応するフォック行列要素 F (I , L) および F (J , L) および F (K , L) の少なくとも一部を、前記ホスト計算機へ送信し、

さらに、前記プロセッサエレメントは、

前記ジョブの終了時に、

前記ブロック I B に含まれる前記縮約基底関数 I 、および前記ブロック J B に含まれる前記縮約基底関数 J 、および前記ブロック K B に含まれる前記縮約基底関数 K 、に対応するフォック行列要素 F (I , J) および F

(I , K) および F (J , K) の少なくとも一部を、前記ホスト計算機へ送信することを特徴とする分子軌道計算方法。

【請求項 8】請求項 6 に記載の分子軌道計算方法において、

前記プロセッサエレメントの処理における前記インデックス L B の切り替わり時に、前記ホスト計算機から前記プロセッサエレメントへ送信されるのは、

前記ブロック K B に含まれる前記縮約基底関数 K および前記ブロック L B に含まれる前記縮約基底関数 L に対応する密度行列要素 P (K , L) のうち、全ての 2 電子積分関数 g の関数値 $g(i, j, k, l)$ の計算が不必要であると判断されない前記縮約基底関数 K , L の組み合わせに対応する行列要素 P (K , L) であることを特徴とする分子軌道計算方法。

【請求項 9】請求項 6 に記載の分子軌道計算方法において、

前記プロセッサエレメントの処理における前記インデックス L B の切り替わり時に、前記ホスト計算機から前記プロセッサエレメントへ送信されるのは、

前記ブロック I B に含まれる前記縮約基底関数 I および前記ブロック J B に含まれる前記縮約基底関数 J および前記ブロック K B に含まれる前記縮約基底関数 K および前記ブロック L B に含まれる前記縮約基底関数 L に対応する密度行列要素 P (K , L) のうち、全ての 2 電子積分関数 g の関数値 $g(i, j, k, l)$ の計算が不必要であると判断されない前記縮約基底関数 K , L の組み合わせに対応する前記密度行列要素 P (K , L) と、

前記ブロック I B に含まれる前記縮約基底関数 I および前記ブロック L B に含まれる前記縮約基底関数 L に対応する前記密度行列要素 P (I , L) と、

前記ブロック J B に含まれる前記縮約基底関数 J および前記ブロック L B に含まれる前記縮約基底関数 L に対応

する前記密度行列要素 P (J , L) であることを特徴とする分子軌道計算方法。

【請求項 10】請求項 7 に記載の分子軌道計算方法において、

前記プロセッサエレメントの処理における前記インデックス L B の切り替わり時に、前記ホスト計算機からプロセッサへ送信されるのは、

前記ブロック I B に含まれる前記縮約基底関数 I および前記ブロック J B に含まれる前記縮約基底関数 J および前記ブロック K B に含まれる前記縮約基底関数 K および前記ブロック L B に含まれる前記縮約基底関数 L に対応する密度行列要素 P (I , L) および P (J , L) および P (K , L) のうち、

全ての 2 電子積分関数 g の関数値 $g(i, j, k, l)$ の計算が不必要であると判断されないか、あるいは前記ブロック J B 内の全ての前記縮約基底関数 J と L との組み合わせで全ての 2 電子積分関数 g の関数値 $g(i, k, j, l)$ の計算が不必要であると判断されないか、あるいは前記ブロック I B 内の全ての前記縮約基底関数 I と L との組み合わせで全ての 2 電子積分関数 g の関数値 $g(i, l, j, k)$ の計算が不必要であると判断されない前記密度行列要素 P (K , L) と、

全ての 2 電子積分関数 g の関数値 $g(i, l, j, k)$ の計算が不必要であると判断されないか、あるいは前記ブロック J B 内の全ての前記縮約基底関数 J と L との組み合わせで全ての 2 電子積分関数 g の関数値 $g(i, k, j, l)$ の計算が不必要であると判断されないか、あるいは前記ブロック K B 内の全ての前記縮約基底関数 K と L との組み合わせで全ての 2 電子積分関数 g の関数値 $g(i, j, k, l)$ の計算が不必要であると判断されない前記密度行列要素 P (I , L) と、

全ての 2 電子積分関数 g の関数値 $g(i, k, j, l)$ の計算が不必要であると判断されないか、あるいは前記ブロック I B 内の全ての前記縮約基底関数 I と L との組み合わせで全ての 2 電子積分関数 g の関数値 $g(i, l, j, k)$ の計算が不必要であると判断されないか、あるいは前記ブロック K B 内の全ての前記縮約基底関数 K と L との組み合わせで全ての 2 電子積分関数 g の関数値 $g(i, j, k, l)$ の計算が不必要であると判断されない前記密度行列要素 P (J , L) であることを特徴とする分子軌道計算方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】この発明は、大規模で、特定の対称性を有する行列要素計算、特に、非経験的分子軌道法を用いた分子シミュレーションにおいてフォック行列要素計算を高速に処理するために用いて好適な、行列要素の並列計算方法に関する。

【0002】

【従来の技術】化学の分野において、分子の状態や挙動

を数値的に解析する手法として、分子軌道法、分子動力学法、モンテカルロ法などがある。その中でも、非経験的分子軌道計算法は、第一原理に基づいた量子力学的計算で、分子中の電子の挙動を記述することを目的としている。そのため、この手法は、分子シミュレーションの基盤として位置づけられ、物質構造や化学反応の詳細な解析に用いられている工業的に重要な手法である。

【0003】非経験的分子軌道計算法では、分子を構成する原子の原子核と軌道電子との距離の2乗に経験的な定数を乗じたものを指数とする指数関数の逆数あるいは

【0004】分子軌道における基底関数の線形結合係数を決めることが、非経験的分子軌道計算法での主要な処理であるが、その計算には、基底関数の数の4乗に比例した計算量と記憶容量を必要とする。そのため、非経験的分子軌道計算法は、現状では、100原子程度の規模の分子系に適用されているに過ぎない。生命現象／化学現象の分子論的解明を、より現実的なものとするためには、数1000原子規模の分子系への適用も視野に入れた、非経験的分子軌道計算専用の計算システムの開発が必須である。

【0005】[非経験的分子起動法の概要] 非経験的分子軌道計算法では、分子の状態 Ψ を、分子中の電子の空間的な軌道に相当する電子軌道関数 ϕ_μ を用いて記述する。ここで μ は複数ある分子軌道の μ 番目という意味の添え字である。分子軌道 ϕ_μ は、原子軌道 χ_i の線形結合で、図13の(数式1)のように近似的に表わされる。

【0006】ここで、(数式1)において、 I は複数ある原子軌道の I 番目という意味の添え字である。なお、原子軌道は基底関数とも呼ばれることがある。この明細書中では、以降、原子軌道のことを基底関数と呼ぶ。また、数式1に現れる $C_{I\mu}$ は、線形結合係数である。数式1における I に関する総和は、計算の対象とする分子を構成する全ての基底関数に関するものである。

【0007】さて、量子力学的に分子軌道を記述するためには、良く知られるパウリの排他律を、分子内の電子の状態が満たさなければならない。電子のスピンを考慮に入れて、パウリの排他律を満たすように、 $2n$ 電子系の分子の状態 Ψ を記述する表式として、図13の(数式2)のようなスレーター行列式が用いられる。ここで、(数式2)において、 $\alpha(x)$ および $\beta(x)$ は、 x 番目の電子のスピンが、それぞれ、上向きおよび下向きの状態を表わしている。

【0008】 $2n$ 電子系に対するハミルトニアン H は、1電子部分 H_1 と2電子部分 H_2 との和という形式で、図13の(数式3)から(数式5)のように書き表され

る。

【0009】図13の(数式4)において、右辺の $(\cdot \cdot \cdot)$ 内の第1項は、電子 p の運動エネルギー、第2項は p 番目の電子と A 番目の原子核との相互作用である。

(数式4)において、 Σ_i (この明細書で Σ_i は、 i についての総和を取ることを表すものとする。以下、同じ)は全電子に関する総和、 Σ_A は全原子核に関する総和、 Z_A は原子核 A の電荷、 r_{pA} は電子 p と原子核 A との距離である。

【0010】また、図13の(数式5)は、電子 p と電子 q との間の相互作用を表わしており、 $\Sigma_p \Sigma_q (\cdot \cdot \cdot)$ は2個の電子の組み合わせに関する総和、 r_{pq} は電子 p 、 q 間の距離である。

【0011】上記のハミルトニアン H と、(数式2)のスレーター行列式とを用いると、分子エネルギーの期待値 ϵ が、図14の(数式6)～(数式9)のように表わされる。

【0012】(数式6)において、 Σ_μ および Σ_ν は、 n 個(n は正の整数)ある分子軌道に関する総和である。(数式7)は「コア積分」と呼ばれ、代表として番号1の電子について書かれている。また、(数式8)および(数式9)は、それぞれ「クーロン積分」および「交換積分」と呼ばれ、代表として電子1および電子2について書かれている。

【0013】(数式6)を基底関数を用いて書き直すと、図15に示す(数式10)～(数式13)に示すようなものになる。(数式13)で表わされる積分を、「2電子間反発積分」あるいは省略して「2電子積分」と呼ぶ。

【0014】(数式10)で表わされる分子エネルギーの期待値 ϵ は、 $C_{I\mu}$ という未知数を含んでおり、このままでは数値が得られない。 $C_{I\mu}$ は、(数式1)における線形結合定数であり、 μ は1から n (分子軌道の数)の整数、 I は1から N (N が基底関数の数であり、正の整数)の整数である。以下では、 $C_{I\mu}$ を要素とする $N \times n$ 行列 C を係数行列と呼ぶ。

【0015】期待値 ϵ が最小となるように係数行列を決定し、基底状態の波動関数 Ψ を求める手法の1つとして、ハートリー・フォック・ローサーンの変分法(以下、HFR法と略称する)が用いられる。導出過程は省略し、HFR法の結果として得られる式を、図16の(数式14)～(数式18)に示す。

【0016】 F_{IJ} はフォック行列要素、 P_{KL} は密度行列要素と、それぞれ呼ばれる。以下の説明では、これらを $F(I, J)$ 、 $P(K, L)$ のように表記する場合がある。これらは、1から N の値をとる各 I, J, K, L に対して数値を持っており、それぞれ $N \times N$ 行列の形で表わされる。

【0017】(数式14)を解くことにより、係数行列が求まる。(数式14)は、1から n の間の全ての μ 、

10

20

30

40

50

および 1 から N の間の全ての I に対して存在するので、 $n \times N$ 本の連立方程式になっている。

【0018】(数式 14) を解いて得られた係数行列 C の計算には、密度行列 P が用いられている。密度行列 P は、(数式 18) に示すように係数行列 C から計算される。そのため、具体的な計算手順としては、まず、適当に係数行列 C を与えておき、それを用いて計算した密度行列 P を使って、(数式 15) でフォック行列 F を計算し、(数式 14) の連立方程式を解いて新たな係数行列 C を得る。密度行列 P の元となる C と、結果として得られる C との間の差が十分小さく、すなわち自己無撞着になるまで、上記の計算を繰り返し行う。この反復計算を自己無撞着計算 (以下、SCF 計算と称する) と呼ぶ。

【0019】実際の計算で最も時間を要するのは、(数式 15) のフォック行列要素 F_{ij} の計算である。これは、全ての I, J に対して、この (数式 15) を計算しなければならないこと、および各 I, J の組み合わせに対して、密度行列要素 P_{kl} の K, L に関する和を計算しなければならないことに起因する。

【0020】SCF 計算の手法には 2 通りある。1 つはディスクストレージ SCF 法と呼ばれる手法で、1 回目の SCF 計算の際に得た 2 電子積分の値を全てディスクに保存しておき、2 回目以降は必要な 2 電子積分をディスクから取り出して用いる手法である。もう 1 つはダイレクト SCF 法と呼ばれる手法で、SCF 計算の度に 2 電子積分の計算をやり直す手法である。

【0021】現在では、ディスク容量の制限やアクセス時間の大きさなどから、後者のダイレクト SCF 法を用いるのが主流である。このダイレクト SCF 法による分子軌道計算においては、SCF 計算の 1 回あたりに、N⁴ にほぼ比例する個数の 2 電子積分の計算を行わなければならないため、2 電子積分計算を高速に行うことが分子軌道計算を高速化することに直結する。

【0022】2 電子積分 $G(I, J, K, L)$ 、密度行列 P (K, L)、およびフォック行列 F (I, J) の対称性に関して、ここで言及しておく。

【0023】2 電子積分は、(数式 13) から明らかのように、図 16 の (数式 19) に示すような対称性を有している。したがって、(数式 19) の内の 1 つに関して数値を得ることができれば、他の 7 つについても数値が得られたことになる。

【0024】また、図 16 の (数式 18) から、 $P(K, L) = P(L, K)$ であることがわかり、図 16 の (数式 15) および図 15 の (数式 11) から、 $F(I, J) = F(J, I)$ であることがわかる。

【0025】[縮約基底関数と原始基底関数] 非経験的分子軌道法では、図 17 の (数式 20) に示すような基底関数が一般的に用いられる。この (数式 20) におい

て、r, n, R はベクトルであり、添え字 x, y, z の付いたものがその成分である。r は電子の座標、n は電子の角運動量、R は原子核の座標である。

【0026】 $n_x + n_y + n_z = \lambda$ は、角運動量の大きさであり、軌道量子数とも呼ばれる。この軌道量子数 λ が、0 の場合にその軌道を s 軌道、1 の場合にその軌道を p 軌道、2 の場合にその軌道を d 軌道などと呼ぶ。

【0027】 ζ_m は軌道指数であり、軌道の空間的な広がり具合を示す。軌道指数の異なる複数の軌道の線形結合で 1 つの基底関数を表わす場合があり、そのようにして表わした基底関数を縮約基底関数と呼び、線形結合係数 d_m を縮約係数と呼ぶ。これに対して、線形結合される前の、図 17 の (数式 21) の形の関数 ψ を原始基底関数と呼ぶ。

【0028】縮約基底関数 χ は、I, J, K, L のように大文字で番号付けをし、また、原始基底関数 ψ は、i, j, k, l のように小文字で番号付けするのが慣例であり、本明細書中でもこれに従う。

【0029】[縮約シェルと原始シェル] 軌道量子数が 1 の場合の縮約基底関数には、 $n = (1, 0, 0)$ の場合、 $n = (0, 1, 0)$ の場合、 $n = (0, 0, 1)$ の場合の 3 通りが存在する。同様に、軌道量子数が 2 の場合には 6 通り (あるいは、基底関数の構成の仕方によっては 5 通り) の縮約基底関数が存在する。

【0030】(数式 20) のうちの図 17 の (数式 22) で示す部分が共通な、これら複数の縮約基底関数の集合を、縮約シェルと呼ぶ。p 軌道の縮約シェルは 3 つの縮約基底関数で構成され、また、d 軌道の縮約シェルは 6 つ (または 5 つ) の縮約基底関数で構成される。s 軌道の場合にも、便宜上 1 つの縮約基底関数の集合を縮約シェルと呼ぶ。

【0031】(数式 21) のうちの $\exp[-\zeta(r-R)^2]$ の部分が共通な、原始基底関数の集合を、同様に原始シェルと呼ぶ。縮約シェルは、R, S, T, U のように大文字で番号付けをし、原始シェルは、r, s, t, u のように小文字で番号付けするのが慣例であり、本明細書中でもこれに従う。

【0032】分子軌道計算の実施に際しては、計算の対象とする分子を構成する原子毎に軌道量子数の異なる複数の縮約シェルを用意し、それら全ての集合を基底関数のセットとして用いる。原子核座標 R と軌道量子数 λ との組み合わせ (R, λ) で、1 つの縮約シェルを表わすことができる。

【0033】[2 電子積分の表式] 縮約基底関数で表わされる 2 電子積分 $G(I, J, K, L)$ は、原始基底関数を用いると、図 17 の (数式 23) のように表わされる。ここで、 $g(i, j, k, l)$ は、図 17 の (数式 24) のように表すことができる。

【0034】 $G(I, J, K, L)$ を、縮約基底関数で表現した 2 電子積分と呼び、 $g(i, j, k, l)$ を、

原始基底関数で表現した 2 電子積分と呼ぶが、以降の説明では、どちらも単に 2 電子積分と呼ぶ場合がある。 $g(i, j, k, l)$ も、図 17 の (数式 25) で示すような対称性を有している。

【0035】さて、原始基底関数 ψ は、その角運動量 n 、軌道指数 l 、原子核座標 R の組み合わせで、一意的に示することができる。 i, j, k, l 番目の原始基底関数が、図9に示す表1のような角運動量、軌道指数、原子核座標を有するものと仮定する。

【0036】説明の煩雑さを避けるために、以下の説明 10
では、原始基底関数の番号 i, j, k, l の代わりに、
それぞれの角運動量 a, b, c, d を用いて 2 電子積分
を $[a b, c d]$ のように表わすことにする。

【0037】上記のように用意された基底関数セットを用いて2電子積分を計算する効率的な手法を、文献1 (S. Obara and A. Saika, JCP, vol. 84, no. 7, p. 3964, 1986) に従って説明する。

【0038】まず、 a, b, c, d が全て s 軌道、すなわち $a=0_s=(0, 0, 0)$, $b=0_s=(0, 0, 0)$, $c=0_s=(0, 0, 0)$, $d=0_s=(0, 0, 0)$ である場合には、(数式24)の2電子積分は、図18の(数式26)～(数式34)に示すように求まる。

【0039】ここで、(数式26)に現れる $[\cdot, \cdot, \cdot]$ 「 \cdot 」は補助積分、 m は補助インデックスであるが、これらについては後で述べる。(数式27)の積分範囲は、0から1である。

【0040】また、 a 、 b 、 c 、 d のうち1つでも s 軌道以外のものがある場合には、図19の(数式35)および(数式36)に示す漸化式を用いて計算する。

【0041】(数式35)で、添え字の*i*は、*x*または*y*または*z*成分であることを示す。また、 1_i は、*i*成分のみ1で、他は0であるようなベクトルである。さらに、 $N_i(n)$ は、角運動量*n*の*i*成分の値を示すものである。(数式35)は、左辺の補助積分に現れる角運動量の1つは右辺では確実に1以上減少し、また、左辺の補助積分の補助インデックスは右辺では同じあるいは1だけ増加する、という性質を有している。

【0042】補助積分 $[\cdot\cdot\cdot, \cdot\cdot\cdot]^{(s)}$ は、補助イン
 デックス m が 0 であるときに、2 電子積分 $[\cdot\cdot\cdot, \cdot\cdot\cdot]$ に正確に一致するものであり、2 電子積分の計算を補助するものである。どんなに角運動量の大きな基底関数を含んだ 2 電子積分であっても、(数式 35) を繰り返し用いて角運動量を減少させ、最後には、全て角運動量が $(0, 0, 0)$ であるような補助積分に行き着くことができる。角運動量が $(0, 0, 0)$ であるような補助積分は、(数式 26) を用いて計算できるので、その数値と適当な係数を乗じ、加算すれば 2 電子積分の数値が得られる。

【0043】実際の計算は、以下のように行う。まず、
(数式35)に従って、2電子積分を8つ以下の補助積
分を用いた形式に表す。ここで現れた補助積分に対し
て、さらに、(数式35)を適用する。このような手続
きを繰り返して、全て角運動量が(0, 0, 0)である
ような補助積分に行き着くまでの道筋を、計算手順とし
て記録しておく。

【0044】次に、(数式26)を用いて角運動量が(0, 0, 0)であるような補助積分を計算し、そこから出発して、先ほどの計算手順をたどりながら補助積分の数値を計算していき、最後に目的とする2電子積分の数値を得る。

【0045】(数式35)が有するもう一つの重要な性質は、2電子積分に現れる4つの角運動量の組み合わせが同じであれば、軌道指数や原子核座標の組み合わせが異なっても、上記の計算手順としては全く同じものを用いることができることである。計算の実行に際しては、軌道指数や原子核座標に応じて補助積分に乗じる係数を変えてやるだけで良い。

【0046】【カットオフ】上述したように、計算しなければならない縮約基底関数で表わした2電子積分の数は、縮約基底関数の数Nに対して N^4 となる。実際に数値を得なければならないのは、原始基底関数で表わした2電子積分の方であるが、その総数は、縮約基底関数で表わした2電子積分の数の数倍から数10倍（縮約基底関数を構成する原始基底関数の数、すなわち縮約数に依存する）に及ぶ。

【0047】この個数を減らす手法として、第1に考えられるのは、(数式19)あるいは(数式25)に記した対称性を利用することである。しかしながら、この方法では最も効率化を行っても2電子積分の計算量は1/8にしかない。

【0048】もう1つの手法は、計算精度の観点から、不必要と判断できる2電子積分の計算を、積極的に排除する方法である。不必要な2電子積分の判断は、以下のように行うことができる。

【0049】上述したように、全ての2電子積分の数値は、(数式26)に示した全て角運動量が(0, 0, 0)であるような補助積分[00, 00]^(*)の数値に基づいて計算される。したがって、2電子積分の数値の、フォック行列要素の数値への寄与が計算誤差の程度であるかどうかを、[00, 00]^(*)の数値で判断することが可能である。さらに、[00, 00]^(*)の数値の大きさは、(数式29)に示した関数K(と、と', R, R')の値から、さらに、それは、図19の(数式37)の大きさから判断することができる。

【0050】したがって、(と_a、A、と_b、B)の数値の組み合わせで、(数式25)の1つ目の関数Kの大きさを見積ることで、2電子積分[a b, **]を計算する必要があるかどうかを判断し、また、(と_a、C、

と、 D) の数値の組み合わせで、(数式26)の2つ目の関数 K の大きさを見積ること、2電子積分 $[**]$ を計算する必要があるかどうかを判断することができる。

【0051】このようにして、不必要な2電子積分の計算を排除することを、「カットオフする」と呼ぶことにする。上記の例で、 a および b の情報だけから判断してカットオフする場合には、 a と b でのカットオフ、 c および d の情報だけから判断してカットオフする場合には、 c と d でのカットオフ、と呼ぶ場合がある。このように、 a と b だけで、あるいは c と d だけでカットオフするかどうかの判断ができるのは、図19の(数式37)の最大値が1で下限値が0だからである。このようにカットオフを行うことにより、計算しなければならない2電子積分は、概略で N^2 に比例する個数となり、計算量を大幅に低減できる。

【0052】上述のことから、 N が大きい場合には、2電子積分の対称性を利用することによる効果よりも、カットオフによる計算量低減の効果の方が桁違いに大きく、これを取り入れることによって、非経験的分子軌道計算におけるフォック行列の計算に要する処理時間が大きく短縮できることがわかる。

【0053】[分子軌道計算機システムの例] フォック行列要素の計算を、並列計算機を用いて高速に行うシステムの例として、文献2(白川他, "超高速分子軌道計算専用機MOEのアーキテクチャ", 電子情報通信学会技術報告, vol. CPSY96-46, no. 5, pp. 45-50, 1996)に記載のシステムがある。

【0054】この文献2では、ホスト計算機に、複数のプロセッサエレメントをバスを介して接続して並列処理を行うシステムが示されている。この文献2では、このような構成を有する並列処理システムのアーキテクチャの検討に際して、 R , S , T , U の4つのインデックスで構成される4重ループのまわし方および並列化を行う部分の種々の方法に関して、全体の計算量およびプロセッサエレメントに必要なメモリ量を見積っている。

【0055】文献2に記載されている並列処理システムは、個々のプロセッサ・エレメントが高い演算処理能力を有する上、システム全体を低価格で実現することが可能であるため、コストパフォーマンスの優れた計算システムを提供することができる。しかしながら、文献2では、前述したカットオフを考慮する場合の方法や具体的なループの制御方法への言及がなく、効率的な処理が行えるかどうか不明であった。

【0056】[I. Fosterらの方法] フォック行列要素の計算を、並列計算機を用いて効率的に行うアルゴリズムとして、文献3(I. T. Foster, et al., "Toward High-Performance Computational Chemistry: I. Scalable Fock Mat

rix Construction Algorithms", Journal of Computational Chemistry, vol. 17, no. 1, p. 109, 1996)に記載のアルゴリズムがある。

【0057】この文献3では、幾つかのフォック行列要素計算アルゴリズムに関して、その計算量およびホスト計算機と、複数のプロセッサ・エレメントとの間の通信量を解析している。その内容を、以下に説明する。

【0058】第1のアルゴリズムは、カノニカル法と呼ばれる最も簡単なアルゴリズムである。この手法では、1つのプロセッサ・エレメントに、図20に示す(数式38)の関係を満たす4個の縮約基底関数 I , J , K , L と、6個の密度行列要素 P_{IJ} , P_{IK} , P_{IL} , P_{JK} , P_{JL} , P_{KL} とを渡し、プロセッサエレメントには2電子積分を計算させて、フォック行列要素 F_{IJ} , F_{IK} , F_{IL} , F_{JK} , F_{JL} , F_{KL} の一部を、図20の(数式39)に従って計算させる。

【0059】1つの2電子積分を計算する間に、ホスト計算機とプロセッサエレメントとの間で通信される行列要素の数を、perERIという単位で勘定することとすると、この場合には、通信データ数が12[perERI]となる。

【0060】第2のアルゴリズムは、トリブルソート法と呼ばれるアルゴリズムである。図21の(数式40)の関係を満たす4個の縮約基底関数 I , J , K , L と、6個の密度行列要素 P_{IJ} , P_{IK} , P_{IL} , P_{JK} , P_{JL} , P_{KL} とを、1つのプロセッサエレメントに渡し、そのプロセッサエレメントに3つの2電子積分 $G(I, J, K, L)$, $G(I, K, J, L)$, $G(I, L, J, K)$ を計算させて、フォック行列要素 F_{IJ} , F_{IK} , F_{IL} , F_{JK} , F_{JL} , F_{KL} の一部を、図21の(数式41)に従って計算させる。

【0061】この場合、3つの2電子積分を計算する間に、6つの密度行列要素と、6つのフォック行列要素の転送が必要であるので、通信データ数は、4[perERI]である。したがって、通信データ数の観点からカノニカル法より優れていると言える。

【0062】しかしながら、プロセッサエレメントにおける原始基底関数で表わした2電子積分の計算時間を、例えば2マイクロ秒($=10^{-6}$ 秒、以下では μ sと記す)と仮定し、また、縮約基底関数の平均の縮約数を1.5と仮定し、密度行列要素やフォック行列要素が、倍精度の浮動小数点数すなわち64ビットのデータサイズであると仮定すると、1つの縮約基底関数で表わした2電子積分の計算時間は、約10 μ sとなり、1つのプロセッサエレメント当たりで、25.6Mbps(25.6×10^6 bit per second)の通信性能が、ホスト計算機とプロセッサエレメントとの間で必要とされることになる。

【0063】計算性能を上げるために、プロセッサエレメント数 M を、例えば100とした場合には、全体の通信性能としては、2560Mbpsが要求されることとなるが、現在の技術で、このような通信性能を達成するのは容易でない。

【0064】安価な通信手段、例えばIEEE1394バス規格で定められたシリアル通信では、200Mbpsの通信性能が実現できるが、それを用いてトリブルソート法を採用したフォック行列要素並列計算を行うと、全体の処理時間が通信時間で律速されてしまい、行列の対称性を利用した計算時間低減の効果は得られなくなってしまう。

【0065】第3のアルゴリズムは、単純ブロック法と呼ばれる手法である。これは、さらに、カノニカル法に基づいたものと、トリブルソート法に基づいたものとに細分類できる。この第3のアルゴリズムは、縮約基底関数をブロック化しておくことにより、密度行列要素やフォック行列要素の利用効率を高め、通信量を低減する手法である。

【0066】トリブルソート法に基づいて、その手法を説明する。まず、 N 個ある縮約基底関数を、 I_c 個毎に、 $n (=N \div I_c)$ 個のブロックに分割する。ブロックの番号を、 I_b, J_b, K_b, L_b のように表わすことにする。次に、図21の(数式42)の関係を満たす4個の縮約基底関数ブロック I_b, J_b, K_b, L_b と、6個の密度行列要素ブロック $P(I_b, J_b), P(I_b, K_b), P(I_b, L_b), P(J_b, K_b), P(J_b, L_b), P(K_b, L_b)$ を、1つのプロセッサエレメントに渡す。渡される密度行列要素の数は $6 I_c^2$ 個となる。

【0067】プロセッサエレメントが計算する2電子積分は、 $G(I, J, K, L), G(I, K, J, L), G(I, L, J, K)$ に対応する $3 I_c^4$ 個の2電子積分であり、プロセッサエレメントは、前述の(数式41)と同様に、フォック行列要素ブロック $F(I_b, J_b), F(I_b, K_b), F(I_b, L_b), F(J_b, K_b), F(J_b, L_b), F(K_b, L_b)$ を計算して、ホスト計算機に送り返す。

【0068】このときに送り返されるフォック行列要素数も、 $6 I_c^2$ 個である。その結果、通信データ数は、 $12 I_c^2 \div 3 I_c^4 = 4 / I_c^2$ [perERI]となる。つまり、ブロック内の縮約基底関数の数を多くすればするほど、密度行列要素やフォック行列要素の利用効率が高まり、通信量が低減する。なお、カノニカル法の場合には、図21の(数式42)に代わり(数式43)を用いる。

【0069】さらに、通信量を低減する第4の手法として、列ブロック法がある。これは、単純ブロック法において、 I_b, J_b, K_b の組み合わせが同じで、 L_b だけが異なる計算を、全て1つのプロセッサエレメントに

割り当てる手法である。この手法のフローチャートを図10に示した。なお、図10において破線で示した矢印の部分は、その後の処理がその前の処理で律速されるのではなく、他の処理系からの情報の入力待ちとなることを示している。

【0070】列ブロック法も、さらに、カノニカル法に基づいたものとトリブルソート法に基づいたものとに細分類できるが、トリブルソート法に基づいて、図10のフローチャートを参照しながら説明する。

【0071】まず、最初に単純ブロック法と同様に、 N 個ある縮約基底関数を、 I_c 個毎に $n (=N \div I_c)$ 個のブロックに分割する(ステップS1)。次に、ホスト計算機は、特定のプロセッサエレメントに割り当てる縮約基底関数ブロック I_b, J_b, K_b の組み合わせ(I_b, J_b, K_b)を決定する(ステップS2)。

【0072】次に、ホスト計算機は、プロセッサエレメントへ、前述の(数式42)の関係を満たす3個の縮約基底関数ブロック I_b, J_b, K_b に対応して、要素数が各々 $I_c \times I_c$ 個の密度行列要素ブロック $P(I_b, J_b), P(I_b, K_b), P(J_b, K_b)$ および要素数が各々 $K_b \times I_c \times I_c$ 個の密度行列要素ブロックの列 $P(I_b, L), P(J_b, L), P(K_b, L)$ を送信する(ステップS3およびステップS4)。但し、 L は、1から $K_b \times I_c$ の範囲の全てである。

【0073】次に、これらをステップS11およびステップS12で受信したプロセッサエレメントは、内部で L に関するループをまわし、単純ブロック法と同じ2電子積分およびフォック行列要素を計算する(ステップS13)。全ての L に対する計算が終了すると、プロセッサエレメントは、要素数が各々 $K_b \times I_c \times I_c$ 個のフォック行列要素ブロックの列 $F(I_b, L), F(J_b, L), F(K_b, L)$ および要素数が各々 $I_c \times I_c$ 個のフォック行列要素ブロック $F(I_b, J_b), F(I_b, K_b), F(J_b, K_b)$ をホスト計算機に送り返す(ステップS14およびステップS15)。

【0074】ホスト計算機は、ステップS5およびステップS6で、それらを受信する。そして、ステップS2に戻り、以上の処理を繰り返す。

【0075】このようにすることにより、密度行列要素やフォック行列要素の利用効率がさらに高まり、2電子積分1つあたりの通信データ数は、 $2 / N I_c + 2 / I_c^2$ [perERI]となり、 $N \gg I_c$ を仮定すれば、単純ブロック法の約半分となる。

【0076】さらに、プロセッサエレメント上の I_b, J_b, K_b の組み合わせを更新した際に、変化するのが K_b のみである場合には、行列要素ブロック $P(I_b, J_b), F(I_b, J_b)$ および行列要素ブロックの列 $P(I_b, L), P(J_b, L), F(I_b, L), F(J_b, L)$ は、プロセッサエレメント上に残して再利

用できるので、通信データ数はさらに減って、 $4/3 N I_c + 2/3 I_c^2$ [perERI] となる。

【0077】列ブロック法と同様の思想で、さらに密度行列要素やフォック行列要素の利用効率を高める第5の手法として、クラスタリング法が、文献2では紹介されている。しかしながら、この手法は、負荷分散あるいはスケーラビリティの観点から劣る手法であるとされており、ここでは説明を省略する。

【0078】

【発明が解決しようとする課題】 [カットオフを考慮した場合の問題点] 上述した文献3に記載されているうちで、最も優れた第4のアルゴリズムであっても、カットオフを考慮した場合には不都合が生じる場合がある。そのような例を以下に示す。なお、カットオフにより生き残る割合を α 、プロセッサエレメント数をM、1つの2電子積分G(I, J, K, L)当たりの計算時間をTerri(μs)、行列要素のデータ長を64ビットとする。

【0079】カノニカル法の場合には、1つのプロセッサエレメントに割り当てられたジョブ当たりに発生する通信量は、最低(すなわち、組み合わせの更新時に変化するのが K_0 だけの場合)でも、

$2(2I_c^2 + K_0 I_c^2) \times 64$ (bit)
となる一方、その間にプロセッサエレメントで計算される2電子積分の個数は、
 $\alpha^2 K_0 I_c^4$ (個)

なので、全体で必要な通信性能は、図22に示す(数式44)のようなものとなる。

【0080】トリプルソート法の場合には、1つのプロセッサエレメントに割り当てられたジョブ当たりに発生する通信量は、最低(すなわち、組み合わせの更新時に変化するのが K_0 だけの場合)でも、

$2(2I_c^2 + K_0 I_c^2) \times 64$ (bit)
となる一方、その間にプロセッサエレメントで計算される2電子積分の個数は、
 $3\alpha^2 K_0 I_c^4$ (個)

なので、全体で必要な通信性能は、図22に示す(数式45)のようなものとなる。

【0081】並列処理のプロセッサ・エレメント数Mを100、プロセッサエレメントにおける原始基底関数で表わした2電子積分 $g(i, j, k, l)$ の計算時間が $2\mu s$ 、縮約基底関数の平均縮約数を1.5(従って、2電子積分G(I, J, K, L)1つ当たりの計算時間Terriを $10\mu s$)、カットオフにより生き残る割合 α を0.05、とした場合に、必要とされるホスト計算機とプロセッサエレメントとの間の通信性能のブロックサイズ I_c に対する依存性を計算すると、カノニカルの場合には、図11に示すように、トリプルソートの場合には、図12に示すようになる。

【0082】カノニカルの場合にも、また、トリプルソ

ートの場合にも、必要な通信性能は K_0 の値に依存して変化するが、 $K_0 > 100$ ではその変化量が小さく、実際の通信性能に応じて、ブロックサイズ I_c を、カノニカルの場合なら、例えば50に、トリプルソートの場合であれば、例えば30に、それぞれ設定することが可能である。

【0083】文献3で前提としているような、十分に大きなデータ保持容量を有するワークステーションを、多数台用いるシステムでは、このような方法で計算を行うことが可能であるが、そのようなシステムを構成するには、多大な費用が必要である。

【0084】一方、文献2にあるような、せいぜい数10Mビット程度の小さな容量のメモリが接続される安価な専用プロセッサ・エレメントを用いた計算システムでは、列単位で保持できる行列要素は10列以下、すなわち、許されるブロックサイズは2から3程度までである。

【0085】この場合には、図11あるいは図12の結果から、通信の手段としてIEEE1394バス規格で定められているような安価なシリアル通信を用いた場合には、その性能が200Mbpsであることを考慮すると、通信性能が律速しない効率的な処理を行うことが不可能である。

【0086】この発明は、以上のような点にかんがみ、安価な通信手段と小容量のメモリを有する多数のプロセッサ・エレメントを用いた並列計算によっても、ホスト計算機とプロセッサ・エレメントとの間の通信性能に律速されることなく、効率的に行列要素計算を行える並列処理方法を提供することを目的とする。

【0087】

【課題を解決するための手段】 上記目的を達成するために、請求項1に記載の発明の行列要素の並列計算方法は、同じ1からNの範囲にある4つの整数インデックスI, J, K, Lを用いて表わされ、 $G(I, J, K, L) = G(I, J, L, K) = G(J, I, K, L) = G(J, I, L, K) = G(K, L, I, J) = G(K, L, J, I) = G(L, K, I, J) = G(L, K, J, I)$ なる関係を満たす関数Gの関数値 $G(I, J, K, L)$ と；2つの前記整数インデックスK, Lを用いて表わされ、 $P(K, L) = P(L, K)$ なる関係を満たす行列Pの要素 $P(K, L)$ と；係数A1と；の積 $A1 \cdot P(K, L) \cdot G(I, J, K, L)$ についての前記範囲の全ての前記Kおよび前記Lに関する総和 $F1(I, J)$ と、前記関数値 $G(I, L, K, J)$ と；前記行列要素 $P(K, L)$ と；係数A2と；の積 $A2 \cdot P(K, L) \cdot G(I, L, K, J)$ に関する前記範囲の全ての前記Kおよび前記Lにおける総和 $F2(I, J)$ との和 $F(I, J) = F1(I, J) + F2(I, J)$ を要素とする行列Fの全要素を、ホスト計算機および複数のプロセッサエレメントにより構成される並列計

算装置を用いて計算する行列要素の並列計算方法において、前記ホスト計算機で前記 1 から N の範囲にある前記インデックスを複数のブロックに分割して、第 1 のブロック群を形成し、前記ホスト計算機で前記第 1 のブロック群のブロックを指示する 3 つの整数インデックス I B, J B, K B に関する所定の 3 重ループを形成し、前記ホスト計算機で前記ブロックを指示する前記 3 つの整数インデックス I B, J B, K B で指定されたものを単位として 1 つのジョブを形成し、前記ホスト計算機で前記 1 つのジョブを、前記並列計算装置内の一つの前記プロセッサエレメントに対して割り当て、前記ホスト計算機は、前記ジョブの割り当てを行う際に、前記ブロック I B に含まれる前記インデックス I、および前記ブロック J B に含まれる前記インデックス J、および前記ブロック K B に含まれる前記インデックス K、に対応する行列要素 $P(I, J)$ および $P(I, K)$ および $P(J, K)$ の少なくとも一部を前記プロセッサエレメントに対して送信し、前記プロセッサエレメントは、前記 1 から N の範囲にある前記整数インデックスを、前記第 1 のブロック群と同じまたは異なる複数のブロックに分割して、第 2 のブロックを形成して、その前記第 2 のブロック群のブロックを指示するインデックス L B に関するループ制御を行うとともに、前記インデックス L B が切り替わる毎に、前記ブロック I B に含まれる前記インデックス I、および前記ブロック J B に含まれる前記インデックス J、および前記ブロック K B に含まれる前記インデックス K、および前記ブロック L B に含まれる前記インデックス L、にそれぞれ対応する行列要素 $P(I, L)$ および $P(J, L)$ および $P(K, L)$ の少なくとも一部を前記ホスト計算機から受信し、前記ブロック I B に含まれる前記インデックス I、および前記ブロック J B に含まれる前記インデックス J、および前記ブロック K B に含まれる前記インデックス K、および前記ブロック L B に含まれる前記インデックス L、にそれぞれ対応する関数値 $G(I, J, K, L)$ の少なくとも一部を計算し、前記計算された関数値 $G(I, J, K, L)$ を用いて、前記ブロック I B に含まれる前記インデックス I、および前記ブロック J B に含まれる前記インデックス J、および前記ブロック K B に含まれる前記インデックス K、および前記ブロック L B に含まれる前記インデックス L、に対応する行列要素 $F(I, J)$ および $F(I, K)$ および $F(J, K)$ および $F(I, L)$ および $F(J, L)$ および $F(K, L)$ の少なくとも一部を所定の計算式に従って計算し、前記ブロック I B に含まれる前記インデックス I、および前記ブロック J B に含まれる前記インデックス J、および前記ブロック K B に含まれる前記インデックス K、および前記ブロック L B に含まれる前記インデックス L、に対応する行列要素 $F(I, L)$ および $F(J, L)$ および $F(K, L)$ の少なくとも一部を前記ホスト計算機へ送信し、さらに、前

記プロセッサエレメントは、前記ジョブの終了時に、前記ブロック I B に含まれる前記インデックス I、および前記ブロック J B に含まれる前記インデックス J、および前記ブロック K B に含まれる前記インデックス K、にそれぞれ対応する行列要素 $F(I, J)$ および $F(I, K)$ および $F(J, K)$ の少なくとも一部を前記ホスト計算機へ送信することを特徴とする。

【0088】また、請求項 2 に記載の発明の行列要素の並列計算方法は、同じ 1 から N の範囲にある 4 つの整数インデックス I, J, K, L を用いて表わされ、 $G(I, J, K, L) = G(I, J, L, K) = G(J, I, K, L) = G(J, I, L, K) = G(K, L, I, J) = G(K, L, J, I) = G(L, K, I, J) = G(L, K, J, I)$ なる関係を満たす関数 G の関数値 $G(I, J, K, L)$ と；2 つの前記整数インデックス K, L を用いて表わされ、 $P(K, L) = P(L, K)$ なる関係を満たす行列 P の要素 $P(K, L)$ と；係数 $A1$ と；の積 $A1 \cdot P(K, L) \cdot G(I, J, K, L)$ についての前記範囲の全ての前記 K および前記 L に関する総和 $F1(I, J)$ と、前記関数値 $G(I, L, K, J)$ と；前記行列要素 $P(K, L)$ と；係数 $A2$ と；の積 $A2 \cdot P(K, L) \cdot G(I, L, K, J)$ に関する前記範囲の全ての前記 K および前記 L における総和 $F2(I, J)$ との和 $F(I, J) = F1(I, J) + F2(I, J)$ を要素とする行列 F の全要素を、ホスト計算機および複数のプロセッサエレメントにより構成される並列計算装置を用いて計算する行列要素の並列計算方法において、前記ホスト計算機で前記 1 から N の範囲にある前記インデックスを複数のブロックに分割して、第 1 のブロック群を形成し、前記ホスト計算機で前記第 1 のブロック群のブロックを指示する 3 つの整数インデックス I B, J B, K B に関する所定の 3 重ループを形成し、前記ホスト計算機で前記ブロックを指示する前記 3 つの整数インデックス I B, J B, K B の 3 つが固定されたものを単位として 1 つのジョブを形成し、前記ホスト計算機で前記 1 つのジョブを前記並列計算装置内の一つのプロセッサエレメントに対して割り当て、前記ホスト計算機は、前記ジョブの割り当てを行う際に、前記ブロック I B に含まれる前記インデックス I、および前記ブロック J B に含まれる前記インデックス J、および前記ブロック K B に含まれる前記インデックス K、に対応する行列要素 $P(I, J)$ および $P(I, K)$ および $P(J, K)$ の少なくとも一部を前記プロセッサエレメントに対して送信し、前記プロセッサエレメントは、前記 1 から N の範囲にある前記インデックスを、前記第 1 のブロック群と同じまたは異なる複数のブロック群に分割して、第 2 のブロック群を形成して、その前記第 2 のブロック群のブロックを指示するインデックス L B に関するループ制御を行うとともに、前記インデックス L B が切り替わる毎に、前記ブロック I

Bに含まれる前記インデックスI、および前記ブロックJBに含まれる前記インデックスJ、および前記ブロックKBに含まれる前記インデックスK、および前記ブロックLBに含まれる前記インデックスL、に対応する行列要素 $P(I, L)$ および $P(J, L)$ および $P(K, L)$ の少なくとも一部を、前記ホスト計算機から受信し、前記ブロックIBに含まれる前記インデックスI、および前記ブロックJBに含まれる前記インデックスJ、および前記ブロックKBに含まれる前記インデックスK、および前記ブロックLBに含まれる前記インデックスL、に対応する関数値 $G(I, J, K, L)$ および $G(I, K, J, L)$ および $G(I, L, J, K)$ の少なくとも一部を計算し、前記計算された関数値を用いて、前記ブロックIBに含まれる前記インデックスI、および前記ブロックJBに含まれる前記インデックスJ、および前記ブロックKBに含まれる前記インデックスK、および前記ブロックLBに含まれる前記インデックスL、に対応する行列要素 $F(I, J)$ および $F(I, K)$ および $F(J, K)$ および $F(I, L)$ および $F(J, L)$ および $F(K, L)$ の少なくとも一部を

所定の計算式に従って計算し、前記ブロックIBに含まれる前記インデックスI、および前記ブロックJBに含まれる前記インデックスJ、および前記ブロックKBに含まれる前記インデックスK、および前記ブロックLBに含まれる前記インデックスL、に対応する行列要素 $F(I, L)$ および $F(J, L)$ および $F(K, L)$ の少なくとも一部を、前記ホスト計算機へ送信し、さらに、前記プロセッサエレメントは、前記ジョブの終了時に、前記ブロックIBに含まれる前記インデックスI、および前記ブロックJBに含まれる前記インデックスJ、および前記ブロックKBに含まれる前記インデックスK、に対応する行列要素 $F(I, J)$ および $F(I, K)$ および $F(J, K)$ の少なくとも一部を、前記ホスト計算機へ送信することを特徴とする。

【0089】また、請求項6に記載の発明の分子軌道計算方法は、それぞれN個(Nは正の整数)の縮約シェルR、S、T、Uのそれぞれに含まれる原始シェルr、s、t、uのそれぞれの成分である原始基底関数i、j、k、lをインデックスとして用いて表わされる2電子積分関数gの関数値 $g(i, j, k, l)$ と；前記原始基底関数kをひとつの構成要素とする縮約基底関数Kおよび前記原始基底関数lをひとつの構成要素とする縮約基底関数Lをインデックスとして用いて表わされる密度行列Pの要素 $P(K, L)$ と；係数A1と；の積 $A1 \cdot P(K, L) \cdot g(i, j, k, l)$ の全ての縮約基底関数に関する総和 $f1(I, J)$ と、前記2電子積分関数gの関数値 $g(i, k, j, l)$ と；前記密度行列Pの前記要素 $P(K, L)$ と；係数A2と；の積 $A2 \cdot P(K, L) \cdot g(i, k, j, l)$ の全ての縮約基底関数に関する総和 $f2(I, J)$ との和 $f(I, J) =$

$f1(I, J) + f2(I, J)$ の、前記原始基底関数i、前記原始基底関数jをそれぞれひとつの構成要素とする縮約基底関数I、縮約基底関数Jに含まれる全ての原始基底関数に関する和で表わされるFock行列の全ての要素 $F(I, J)$ の計算を、ホスト計算機および複数のプロセッサエレメントにより構成される並列計算装置を用いて計算する分子軌道計算方法において、前記ホスト計算機で前記1からNの範囲にある縮約基底関数の前記インデックスを複数のブロックに分割して、第1のブロック群を形成し、前記ホスト計算機で前記第1のブロック群のブロックを指示する3つの整数インデックスIB、JB、KBに関する所定の3重ループを形成し、前記ホスト計算機で前記ブロックを指示する前記3つの整数インデックスIB、JB、KBで指定されたものを単位として1つのジョブを形成し、前記ホスト計算機で前記1つのジョブを、前記並列計算装置内の一つの前記プロセッサエレメントに対して割り当て、前記ホスト計算機は、前記ジョブの割り当てを行う際に、前記ブロックIBに含まれる前記縮約基底関数I、および前記ブロックJBに含まれる前記縮約基底関数J、および前記ブロックKBに含まれる前記縮約基底関数K、に対応する密度行列要素 $P(I, J)$ および $P(I, K)$ および $P(J, K)$ の少なくとも一部を、前記プロセッサエレメントに対して送信し、前記プロセッサエレメントは、前記1からNの範囲にある縮約シェルを、前記第1のブロック群と同じまたは異なる複数のブロックに分割して、第2のブロック群を形成して、その前記第2のブロック群のブロックを指示する整数インデックスLBに関するループ制御を行うとともに、前記インデックスLBが切り替わる毎に、前記ブロックIBに含まれる前記縮約基底関数I、および前記ブロックJBに含まれる前記縮約基底関数J、および前記ブロックKBに含まれる前記縮約基底関数K、および前記ブロックLBに含まれる前記縮約基底関数L、に対応する密度行列要素 $P(I, L)$ および $P(J, L)$ および $P(K, L)$ の少なくとも一部を、前記ホスト計算機から受信し、前記ブロックIBに含まれる前記縮約基底関数Iを構成する前記原始基底関数i、および前記ブロックJBに含まれる前記縮約基底関数Jを構成する前記原始基底関数j、および前記ブロックKBに含まれる前記縮約基底関数Kを構成する前記原始基底関数k、および前記ブロックLBに含まれる前記縮約基底関数Lを構成する前記原始基底関数lに対応する2電子積分の関数値 $g(i, j, k, l)$ の少なくとも一部を計算し、その計算により求められた前記関数値を用いて、前記ブロックIBに含まれる前記縮約基底関数I、および前記ブロックJBに含まれる前記縮約基底関数J、および前記ブロックKBに含まれる前記縮約基底関数K、および前記ブロックLBに含まれる前記縮約基底関数L、に対応するFock行列要素 $F(I, J)$ および $F(I, K)$ および $F(J, K)$ および F

(I, L) および F (J, L) および F (K, L) の少なくとも一部を所定の計算式に従って計算し、前記ブロック I B に含まれる前記縮約基底関数 I、および前記ブロック J B に含まれる前記縮約基底関数 J、および前記ブロック K B に含まれる前記縮約基底関数 K、および前記ブロック L B に含まれる前記縮約基底関数 L、に対応するフォック行列要素 F (I, L) および F (J, L) および F (K, L) の少なくとも一部を前記ホスト計算機へ送信し、さらに、前記プロセッサエレメントは、前記ジョブの終了時に、前記ブロック I B に含まれる前記縮約基底関数 I、および前記ブロック J B に含まれる前記縮約基底関数 J、および前記ブロック K B に含まれる前記縮約基底関数 K、に対応するフォック行列要素 F (I, J) および F (I, K) および F (J, K) の少なくとも一部を前記ホスト計算機へ送信することを特徴とする。

【0090】また、請求項 7 に記載の発明の分子軌道計算方法は、それぞれ N 個 (N は正の整数) の縮約シェル R, S, T, U のそれぞれに含まれる原始シェル r, s, t, u のそれぞれの成分である原始基底関数 i, j, k, l をインデックスとして用いて表わされる 2 電子積分関数 g の関数値 g (i, j, k, l) と；前記原始基底関数 k をひとつの構成要素とする縮約基底関数 K および前記原始基底関数 l をひとつの構成要素とする縮約基底関数 L とをインデックスとして用いて表わされる密度行列 P の要素 P (K, L) と；係数 A 1 との積 A 1 · P (K, L) · g (i, j, k, l) の全ての縮約基底関数に関する総和 f 1 (I, J) と、前記 2 電子積分関数 g の関数値 g (i, k, j, l) と；前記密度行列 P の前記要素 P (K, L) と；係数 A 2 と；の積 A 2 · P (K, L) · g (i, k, j, l) の全ての縮約基底関数に関する総和 f 2 (I, J) との和 f (I, J) = f 1 (I, J) + f 2 (I, J) の、前記原始基底関数 i, 前記原始基底関数 j をそれぞれひとつの構成要素とする縮約基底関数 I, 縮約基底関数 J に含まれる全ての原始基底関数に関する和で表わされる F o c k 行列の全ての要素 F (I, J) の計算を、ホスト計算機および複数のプロセッサエレメントにより構成される並列計算装置を用いて計算する分子軌道計算方法において、前記ホスト計算機で前記 1 から N の範囲にある縮約基底関数の前記インデックスを複数のブロックに分割して、第 1 のブロック群を形成し、前記ホスト計算機で前記第 1 のブロック群のブロックを指示する 3 つの整数インデックス I B, J B, K B に関する所定の 3 重ループを形成し、前記ホスト計算機で前記ブロックを指示する前記 3 つのインデックス I B, J B, K B で指定されたものを単位として 1 つのジョブを形成し、前記ホスト計算機で前記 1 つのジョブを前記並列計算装置内の一つのプロセッサエレメントに対して割り当て、前記ホスト計算機は、前記ジョブの割り当てを行う際に、前記ブロック I B に含

まれる前記縮約基底関数 I、および前記ブロック J B に含まれる前記縮約基底関数 J、および前記ブロック K B に含まれる前記縮約基底関数 K、に対応する密度行列要素 P (I, J) および P (I, K) および P (J, K) の少なくとも一部を、前記プロセッサエレメントに対して送信し、前記プロセッサエレメントは、前記 1 から N の範囲にある縮約シェルを、前記第 1 のブロック群と同じまたは異なる複数のブロック群に分割して、第 2 のブロック群を形成して、その前記第 2 のブロック群のブロックを指示するインデックス L B に関するループ制御を行うとともに、前記インデックス L B が切り替わる毎に、前記ブロック I B に含まれる前記縮約基底関数 I、および前記ブロック J B に含まれる前記縮約基底関数 J、および前記ブロック K B に含まれる前記縮約基底関数 K、および前記ブロック L B に含まれる前記縮約基底関数 L、に対応する密度行列要素 P (I, L) および P (J, L) および P (K, L) の少なくとも一部を、前記ホスト計算機から受信し、前記ブロック I B に含まれる前記縮約基底関数 I を構成する前記原始基底関数 i、および前記ブロック J B に含まれる前記縮約基底関数 J を構成する前記原始基底関数 j、および前記ブロック K B に含まれる前記縮約基底関数 K を構成する前記原始基底関数 k、および前記ブロック L B に含まれる前記縮約基底関数 L を構成する前記原始基底関数 l、に対応する 2 電子積分 g (i, j, k, l) および g (i, k, j, l) および g (i, l, j, k) の少なくとも一部を計算し、この計算された前記関数値を用いて、前記ブロック I B に含まれる前記縮約基底関数 I、および前記ブロック J B に含まれる前記縮約基底関数 J、および前記ブロック K B に含まれる前記縮約基底関数 K、および前記ブロック L B に含まれる前記縮約基底関数 L、に対応するフォック行列要素 F (I, J) および F (I, K) および F (J, K) および F (I, L) および F (J, L) および F (K, L) の少なくとも一部を所定の計算式に従って計算し、前記ブロック I B に含まれる前記縮約基底関数 I、および前記ブロック J B に含まれる前記縮約基底関数 J、および前記ブロック K B に含まれる前記縮約基底関数 K、および前記ブロック L B に含まれる前記縮約基底関数 L、に対応するフォック行列要素 F (I, J) および F (I, K) および F (J, K) の少なくとも一部を、前記ホスト計算機へ送信し、さらに、前記プロセッサエレメントは、前記ジョブの終了時に、前記ブロック I B に含まれる前記縮約基底関数 I、および前記ブロック J B に含まれる前記縮約基底関数 J、および前記ブロック K B に含まれる前記縮約基底関数 K、に対応するフォック行列要素 F (I, J) および F (I, K) および F (J, K) の少なくとも一部を、前記ホスト計算機へ送信することを特徴とする。

【0091】

【作用】上述の構成のこの発明の計算アルゴリズムを用

いと、安価な通信手段と小容量のメモリを有する多数のプロセッサとを用いた並列計算によっても、ホスト計算機とプロセッサとの間の通信性能の影響でシステム全体の処理性能が低下することなく、効率的に行列要素計算を行える並列計算方法および分子軌道計算方法を提供することができる。

【0092】

【発明の実施の形態】以下、この発明の実施の形態を、図を参照しながら説明する。以下に説明する実施の形態では、図1に示すような安価な計算機システムを用いる。

【0093】すなわち、図1は、この実施の形態の並列計算システムの全体の構成を示すもので、ホスト計算機1に対して、バス3を通じて、複数個のプロセッサエレメント2が接続されている。バス3としては、例えばIEEE1394シリアルバスが用いられる。

【0094】なお、各プロセッサエレメント2のメモリ容量は、数10Mビット、例えば20Mビットであり、行列要素が64ビットの浮動小数点数で表わされる密度行列およびフォック行列のサイズが、10000×10000であっても、各行列の10列分は十分格納できる容量のものが用いられる。この程度の容量のメモリを各プロセッサエレメントに持たせることは、現在の技術で十分に可能である。

【0095】この発明の実施の形態の行列要素計算アルゴリズムは、従来の技術の欄で説明した文献3に記載された第3のアルゴリズムである単純ブロック法を改良したものである。

【0096】この実施の形態で用いるアルゴリズムは、上記の単純ブロック法において、ブロック番号 I_0 、 J_0 、 K_0 の組み合わせが同じで、ブロック番号 L_0 だけが異なる計算を、全て1つのプロセッサエレメントに割り当てる手法であるが、後述するように、この手法は、文献3に記載された第4のアルゴリズムである列ブロック法とは異なる手法であって、列ブロック化法と比較して、小さな容量（せいぜい数10Mbit）のメモリが接続される安価な専用プロセッサを用いた、図1のような並列計算システムにおいても比較的大きなブロックサイズが許されるため、安価なシリアル通信を用いても通信性能が律速しない効率的な処理を行うことが可能となるものである。

【0097】また、ブロック番号 L_0 が指示するブロックの単位は、ブロック番号 I_0 、 J_0 、 K_0 が指示するブロックの単位と一致している必要はなく、別のブロックわけを行うことも可能である点で、単純ブロック法とも異なるものである。

【0098】以下の説明では、 I_0 、 J_0 、 K_0 が指示するブロックは、 I_0 個のインデックスから構成され、 L_0 が指示するブロックは、 I_0 （ I_0 とは異なる）個のインデックスから構成されているものとする。

【0099】なお、この実施の形態の計算手法も、さらに、カノニカル法に基づいたものと、トリプルソート法に基づいたものとに細分類できる。

【0100】この発明の実施の形態の計算手法を、図2に示したフローチャートに沿って説明する。なお、図2において破線で示した矢印の部分は、その後の処理が、その前の処理で律速されるのではなく、他の処理系からの情報の入力待ちとなることを示している。

【0101】まず、最初に、ホスト計算機1は、単純ブロック法と同様に、 N 個ある縮約基底関数を、 I_0 個毎に、 n （ $=N \div I_0$ ）個のブロックに分割する（ステップS101）。次に、ホスト計算機1は、特定のプロセッサエレメントに割り当てる縮約基底関数ブロック I_0 、 J_0 、 K_0 の組み合わせ（ I_0 、 J_0 、 K_0 ）を決定し（ステップS102）、その組み合わせ（ I_0 、 J_0 、 K_0 ）を、割り当てられたプロセッサエレメント2に送る（ステップS103）。

【0102】プロセッサエレメント2は、この縮約基底関数ブロック I_0 、 J_0 、 K_0 の組み合わせ（ I_0 、 J_0 、 K_0 ）を、ステップS201で受信する。

【0103】次に、ホスト計算機1は、プロセッサエレメント2に、図21に示した（数式42）または（数式43）の関係を満たす3個の縮約基底関数ブロック I_0 、 J_0 、 K_0 に対応して、要素数が各々 $I_0 \times I_0$ 個の密度行列要素ブロック $P(I_0, J_0)$ 、 $P(I_0, K_0)$ 、 $P(J_0, K_0)$ を送信する（ステップS104）。

【0104】一方、プロセッサエレメント2は、1からブロック K_0 に含まれる最大のインデックスまでを、 I_0 個ごとに m 個のブロックに分割する（ステップS202）。そして、プロセッサエレメント2は、密度行列要素ブロック $P(I_0, J_0)$ 、 $P(I_0, K_0)$ 、 $P(J_0, K_0)$ を受信する（ステップS203）と、内部で、 L_0 に関するループを、1から m までまわす（ステップS204～S209）。

【0105】このとき、 L_0 が切り替わるたびに、プロセッサエレメント2は、ホスト計算機1へ、要素数が各々 $I_0 \times I_0$ 個の密度行列要素ブロック $P(I_0, L_0)$ 、 $P(J_0, L_0)$ 、 $P(K_0, L_0)$ の送信を要求して、それらを受信する（ステップS205）。

【0106】この後、プロセッサエレメント2は、内部で、 L_0 に関するループをまわす（ステップS206）。このステップS206のループにおいては、 L_0 に含まれる全ての L に関わる2電子積分およびフォック行列要素を計算する。

【0107】このステップS206のループが完了して、1つの L_0 に対する計算が終了すると、プロセッサエレメント2は、要素数が各々 $I_0 \times I_0$ 個のフォック行列要素ブロック $F(I_0, L_0)$ 、 $F(J_0, L_0)$ 、 $F(K_0, L_0)$ を、ホスト計算機1に送り返

す (ステップ S 207)。

【0108】このような処理を、前記 (数式 42) あるいは (数式 43) に示す関係を満たす全ての L_i に対して終了したと判断すると (ステップ S 209)、プロセッサエレメント 2 は、要素数が各々 $I_c \times I_r$ 個のフォック行列要素ブロック $F(I_i, J_i)$, $F(I_i, K_i)$, $F(J_i, K_i)$ を、ホスト計算機 1 に送り返す (ステップ S 210)。

【0109】上述したこの発明の実施の形態の計算方法が、従来例における列ブロック法と比較して優れている点は、カットオフを考慮した場合にも、不都合が生じないことである。その例として、通信の手段としては IEEE 1394 規格で定められたシリアル通信を用い、その性能が 200 Mbps であること、プロセッサエレメント 2 の数 M が 100 であること、プロセッサエレメント 2 における原始基底関数で表わした 2 電子積分 $g(i, j, k, l)$ の計算時間が $2 \mu s$ であること、縮約基底関数の平均縮約数が 1.5 であること、を前提とした場合について説明する。

【0110】まず、カノニカル法の場合に付いて説明する。

【0111】 $I_c \times I_r$ 個の要素 P_{il} で構成される密度行列要素ブロック $P(I_i, L_i)$ の要素は、図 20 の (数式 39) の 4 行目の式で用いられる。したがって、ホスト計算機 1 における密度行列要素 P_{il} の要・不要の判断は、以下の条件が成立するかどうかで行う。

【0112】[条件 1] ; ブロック K_i 中の全ての K と L との間でのカットオフ条件

この条件 1 が成立した場合には、密度行列要素ブロック $P(I_i, L_i)$ の要素は全て不要となるため、これをホスト計算機 1 からプロセッサエレメント 2 へ送信する必要がなくなる。逆に、この条件 1 が成立しない場合には、密度行列要素ブロック $P(I_i, L_i)$ の要素は全て必要となるので、これをホスト計算機 1 からプロセッサエレメント 2 へ送信しなければならない。

【0113】以上のことから、1 つの L_i 当たりで、ホスト計算機 1 からプロセッサエレメント 2 へ送信される密度行列要素 $P(I, L)$ の個数の期待値は、図 22 に示す (数式 46) のように見積ることができる。

【0114】密度行列要素ブロック $P(J_i, L_i)$ の要素は、前記 (数式 39) の 2 行目の式で用いられる。したがって、その要・不要の判断も、前記の条件 1 によって行うので、1 つの L_i 当たりで、ホスト計算機 1 からプロセッサエレメント 2 へ送信される密度行列要素 $P(J, L)$ の個数の期待値も、前記 (数式 46) と同じとなる。

【0115】一方、密度行列要素ブロック $P(K_i, L_i)$ の要素は、(数式 39) の 1 行目の式で用いられる。したがって、その要・不要の判断は、

[条件 2] ; L と K との間でのカットオフ条件

が成立するかどうかのみで行う。

【0116】したがって、1 つの L_i 当たりで、ホスト計算機 1 からプロセッサエレメント 2 へ送信される密度行列要素 $P(K, L)$ の個数の期待値は、図 22 の (数式 47) に示すものとなる。

【0117】したがって、1 つのプロセッサエレメント 2 で、 L_i 当たりで発生する通信量は、プロセッサエレメント 2 からホスト計算機 1 へ送り返されるフォック行列要素数まで考慮して、図 23 の (数式 48) に示すものとなる。

【0118】そして、その間にプロセッサエレメント 2 で計算される 2 電子積分の個数は、

$$\alpha^2 I_c^3 \times I_r \quad (\text{個})$$

なので、全体に必要な通信性能は、図 23 の (数式 49) に示すものとなる。

【0119】なお、通信の実施に際して、カットオフ条件を、どこまで詳細に考慮するかによって、必要な通信性能が異なってくる。カットオフ条件を詳細に考慮すると、通信量が減る代わりに、ホスト計算機 1 における処理量が増える。逆に、カットオフ条件を考慮しないと、ホスト計算機 1 における処理量が減る代わりに、通信量が増える。

【0120】必要な通信性能を計算した結果を、カットオフ条件を全く考慮せずに、全ての密度行列要素を送信する場合について図 3 に、カットオフ条件 2 のみを考慮する (すなわち、密度行列要素ブロック $P(I_i, L_i)$ および $P(J_i, L_i)$ に関しては、カットオフ判断を行わずに全て送信する) 場合について図 4 に、カットオフ条件 1、2 を全て考慮する場合について図 5 に、それぞれ示す。ここでは、 $M=100$, $T_{eri}=10 (\mu s)$, $\alpha=0.05$ とした。

【0121】次にトリプルソート法の場合について説明する。

【0122】 $I_c \times I_r$ 個の要素 P_{il} で構成される密度行列要素ブロック $P(I_i, L_i)$ の要素は、前記 (数式 41) の 4 行目の式で用いられる。したがって、ホスト計算機 1 における密度行列要素 P_{il} の要・不要の判断は、以下の 3 条件が同時に成立するかどうかで行う。

【0123】[条件 3] ; ブロック K_i 中の全ての K と L との間でのカットオフ条件

[条件 4] ; ブロック J_i 中の全ての J と L との間でのカットオフ条件

[条件 5] ; I と L との間でカットオフ条件。

【0124】条件 3 から条件 5 の全てが成立した場合には、密度行列要素ブロック $P(I_i, L_i)$ の要素は全て不要となるため、これをホスト計算機からプロセッサエレメントへ送信する必要がなくなる。逆にこれらの条件が 1 つでも成立しない場合には、密度行列要素ブロック $P(I_i, L_i)$ の要素は全て必要となるので、これをホスト計算機からプロセッサエレメントへ送信しな

ればならない。

【0125】以上のことから、1つの L_i 。当たりで、ホスト計算機1からプロセッサエレメント2へ送信される密度行列要素 $P(I, L)$ の個数の期待値は、図23の(数式50)に示すように見積られる。

【0126】密度行列要素 P_{iL} に関しては、以下の条件6から条件8を用いて、また、密度行列要素 P_{iL} に関しては、以下の条件9から条件11を用いて、それぞれカットオフの判断を行うが、1つの L_i 。あたりで、ホスト計算機1からプロセッサエレメント2へ送信される密度行列要素 $P(J, L)$ および $P(K, L)$ の個数の期待値は、前記(数式50)と同じとなる。

【0127】[条件6] ; ブロック K_i 。中の全ての K と L との間でのカットオフ条件

[条件7] ; J と L との間でカットオフ条件

[条件8] ; ブロック I_i 。中の全ての J と L との間でのカットオフ条件

[条件9] ; K と L との間でカットオフ条件

[条件10] ; ブロック J_i 。中の全ての J と L との間でのカットオフ条件

[条件11] ; ブロック I_i 。中の全ての J と L との間でのカットオフ条件。

【0128】したがって、1つのプロセッサエレメント2で、 L_i 。当りに発生する通信量は、プロセッサエレメント2からホスト計算機1へ送り返されるフォック行列要素数まで考慮して、図23の(数式51)に示すものとなる。

【0129】その間にプロセッサエレメント2で計算される2電子積分の個数は、

$$3\alpha^2 I_i^3 \times I_i \quad (\text{個})$$

なので、全体に必要な通信性能は、図23の(数式52)に示すものとなる。

【0130】なお、この場合にも、通信の実施に際して、カットオフ条件を考慮するかどうかによって、必要な通信性能が異なってくる。カットオフ条件を考慮すると、通信量が減る代わりに、ホスト計算機1における処理量が増える。逆に、カットオフ条件を考慮しないと、ホスト計算機1における処理量が減る代わりに、通信量が増える。

【0131】必要な通信性能を計算した結果を、カットオフ条件を全く考慮せずに全ての密度行列要素を送信する場合について図6に、また、カットオフ条件を全て考慮する場合について図7に、それぞれ示す。ここで、カノニカル法の場合と同様に、 $M=100$ 、 $T_{eri}=10(\mu s)$ 、 $\alpha=0.05$ とした。

【0132】以上の結果から、カノニカル法の場合には、最悪でも、ブロックサイズ I_i 。を90以上に設定することで、また、トリプルソート法の場合には、ブロックサイズ I_i 。を50以上に設定することで、必要な通信性能を200Mbps以下とすることが可能である。

【0133】ブロックサイズ I_i 。および I_o 。を100としても、各並列プロセッサエレメント2上に保有しなくてはならない行列要素数は、120000個であり、要素1個あたり64ビットとしても、要求される容量は7.7Mビット程度である。現在の技術であっても、この程度の容量を有するメモリを、各プロセッサエレメント2に接続することは十分に可能である。

【0134】また、(数式49)および(数式52)からわかるように、全体で必要な通信性能は、 I_i 。に無依存である。したがって、プロセッサエレメントに接続したメモリの空き容量を見て、 I_o 。を可変にすることも可能である。カットオフを考慮した通信制御を行う場合には、メモリの空き容量が大きくなる場合があり、そのときに、 I_o 。を大きく設定することで、通信回数を減らすことができる。

【0135】なお、この発明の実施の形態において、トリプルソート法を用いる場合には、通信量が、カットオフ条件を考慮するかしないかには、ほぼ依存しないので、ホスト計算機1でのカットオフ判断は、ホスト計算機1での処理量を小さくできるように行わないことが望ましい。

【0136】以上説明した、この発明の実施の形態の効果を、図8に示したグラフを用いて説明する。図8において、丸印のプロットは、ホスト計算機1とプロセッサエレメント2との間に必要となる通信性能のブロックサイズ依存性を示すものであり、白丸が従来例の列ブロックアルゴリズムにおけるもの、黒丸がこの発明の実施の形態におけるものである。なお、これらは、いずれもトリプルソート法を用いる場合であり、また、この発明の実施の形態におけるものに関しては、通信に際しては、カットオフを考慮しない場合を示した。

【0137】一方、四角印のプロットは、プロセッサエレメント2に必要なメモリ容量のブロックサイズ依存性を示すものであり、白四角が従来例の列ブロックアルゴリズムにおけるもの、黒四角がこの発明の実施の形態におけるものである。

【0138】図1に示したような安価なシステム構成で、ホスト計算機1とプロセッサエレメント2との間のバスの通信速度が200Mbps、プロセッサエレメント2におけるメモリ容量が20Mビットであると仮定すると、ブロックサイズは、上記のプロットの双方が、図8に示した破線の下側になっている領域に設定しなければならない。

【0139】従来例の場合には、丸印のプロットと四角印のプロットとが、同時に破線の下側になる領域がなく、ブロックサイズをどのように選んでも、通信時間によってシステム全体の処理性能が低下するか、あるいは必要なデータをプロセッサエレメント2のメモリに格納できないという不都合が起きる。

【0140】一方、この発明の実施の形態では、プロッ

クサイズを50から170の間に設定すれば、丸印のプロットと四角印のプロットとが、同時に破線の下側になり、通信時間によってシステム全体の処理性能が低下せず、また、プロセッサエレメントのメモリに必要なデータを全て格納できる。

【0141】以上説明した実施の形態は、非経験的分子軌道法を用いた分子シミュレーションにおいて、フォック行列要素計算を高速に行う場合に、この発明を適用した場合であるが、この発明は、このような非経験的分子軌道法に限らず、種々の並列処理アルゴリズムに適用可能であることは、言うまでもない。

【0142】

【発明の効果】以上説明したように、この発明による並列処理方法によれば、安価な通信手段と小容量のメモリを有する多数のプロセッサエレメントとを用いた並列計算システムによっても、ホスト計算機とプロセッサエレメントとの間の通信性能の影響でシステム全体の処理性能が低下するようなことなく、効率的に行列要素計算を行える。

【図面の簡単な説明】

【図1】この発明による並列処理装置の実施の形態のシステム構成を示すブロック図である。

【図2】この発明の実施の形態におけるホスト計算機およびプロセッサエレメントの処理のフローチャートである。

【図3】この発明の実施の形態でカノニカル法を用いた場合であって、カットオフ条件を全く考慮しない場合に必要となる通信性能とブロックサイズとの関係を示す図である。

【図4】この発明の実施の形態でカノニカル法を用いた場合であって、カットオフ条件を一部考慮する場合に必要な通信性能とブロックサイズとの関係を示す図である。

【図5】この発明の実施の形態でカノニカル法を用いた場合であって、カットオフ条件を全て考慮する場合に必要な通信性能とブロックサイズとの関係を示す図である。

【図6】この発明の実施の形態でトリプルソート法を用いた場合であって、カットオフ条件を考慮しない場合に必要となる通信性能とブロックサイズとの関係を示す図である。

【図7】この発明の実施の形態でトリプルソート法を用いた場合であって、カットオフ条件を考慮する場合に必要な通信性能とブロックサイズとの関係を示す図である。

【図8】従来例およびこの発明の実施の形態でトリプルソート法を用いた場合の、ホスト計算機・プロセッサエレメント間に必要な通信性能およびプロセッサエレメントに必要なメモリ容量とブロックサイズとの関係を示す図である。

【図9】原子基底関数と、その角運動量、軌道指数、原子核座標との対応例を示す図である。

【図10】従来例のFosterの列ブロックアルゴリズムのフローチャートである。

【図11】従来例でカノニカル法を用いた場合に必要となる通信性能とブロックサイズとの関係を示す図である。

【図12】従来例でトリプルソート法を用いた場合に必要となる通信性能とブロックサイズとの関係を示す図である。

【図13】非経験的分子軌道計算法の説明に用いる数式を示す図である。

【図14】非経験的分子軌道計算法の説明に用いる数式を示す図である。

【図15】非経験的分子軌道計算法の説明に用いる数式を示す図である。

【図16】非経験的分子軌道計算法の説明に用いる数式を示す図である。

【図17】非経験的分子軌道計算法の説明に用いる数式を示す図である。

【図18】文献1の2電子積分計算法の説明に用いる数式を示す図である。

【図19】文献1の2電子積分計算法の説明に用いる数式（数式35、35）及びカットオフ条件の説明に用いる数式（数式37）を示す図である。

【図20】Fosterのアルゴリズムの説明に用いる数式を示す図である。

【図21】Fosterのアルゴリズムの説明に用いる数式を示す図である。

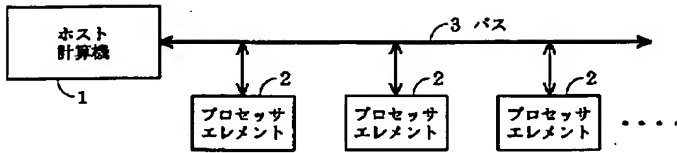
【図22】Fosterのアルゴリズムにおける通信量の説明に用いる数式（数式44、45）およびこの発明の実施の形態における通信量の説明に用いる数式（数式46、47）を示す図である。

【図23】この発明の実施の形態における通信量の説明に用いる数式を示す図である。

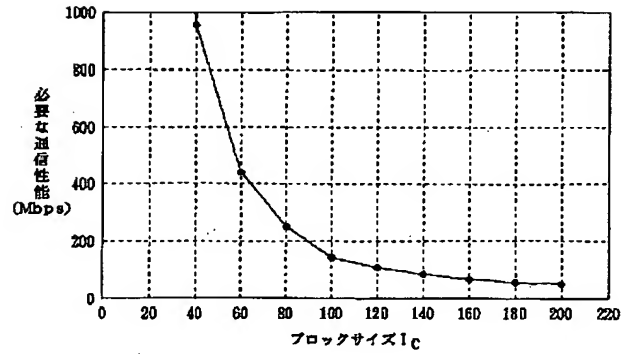
【符号の説明】

- 1 ホスト計算機
- 2 プロセッサエレメント
- 3 バス

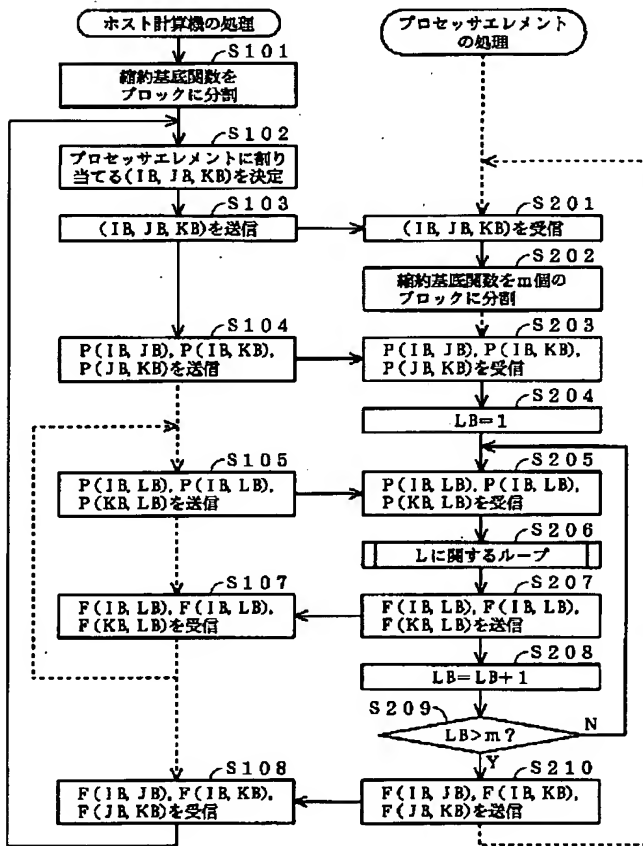
【図1】



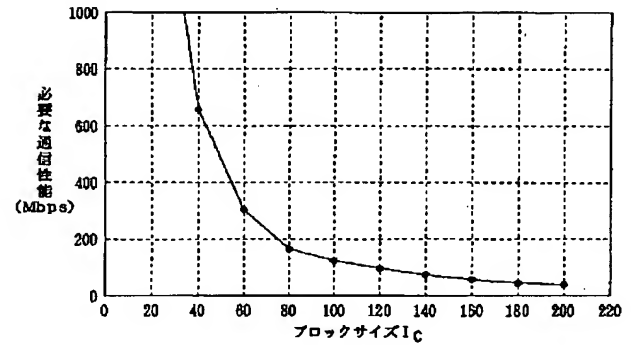
【図3】



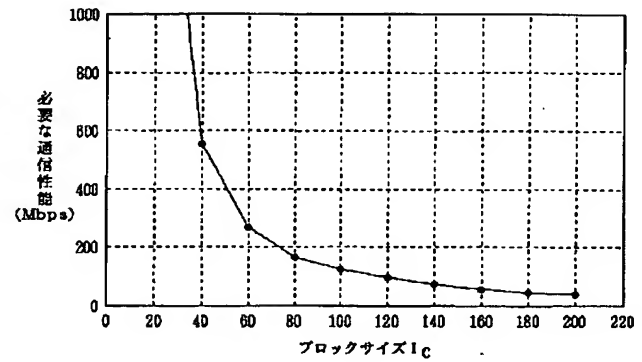
【図2】



【図4】



【図5】

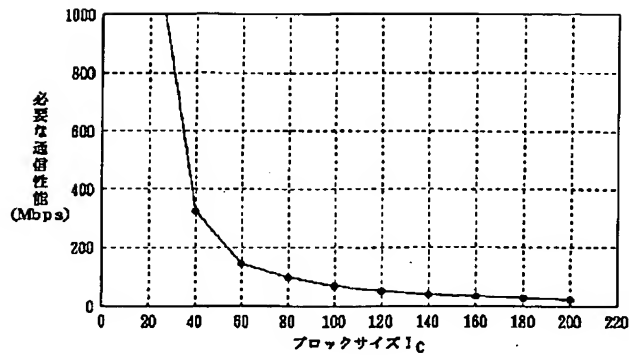


【図9】

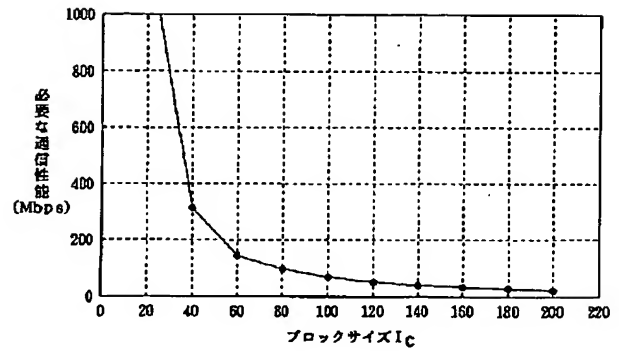
[表1 原始基底関数 i, j, k, l の角運動量、軌道指数、原子核座標]

| 原始基底関数の番号 | 角運動量 | 軌道指数 | 原子核座標 |
|-----------|------|-----------|-------|
| i | a | ζ_a | A |
| j | b | ζ_b | B |
| k | c | ζ_c | C |
| l | d | ζ_d | D |

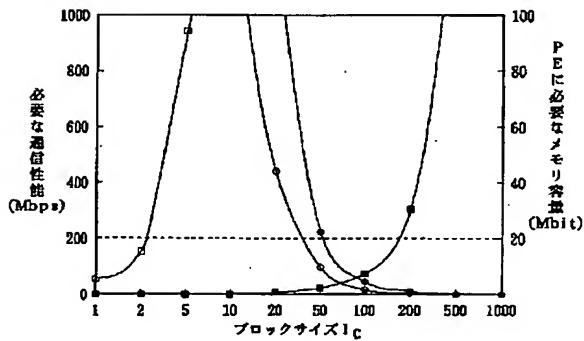
【図6】



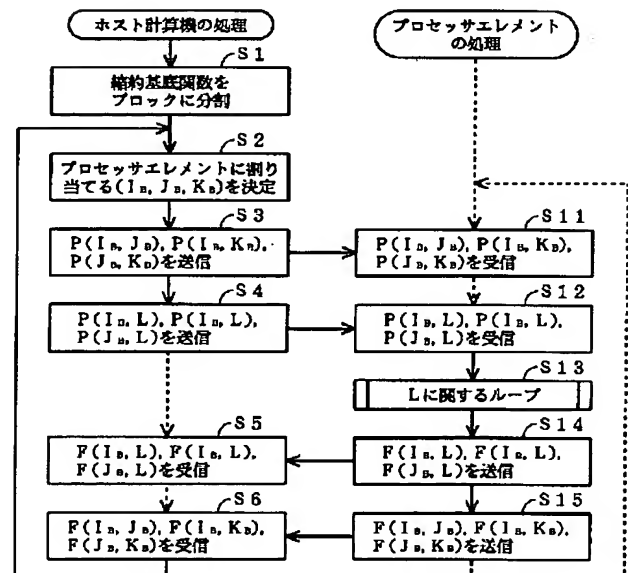
【図7】



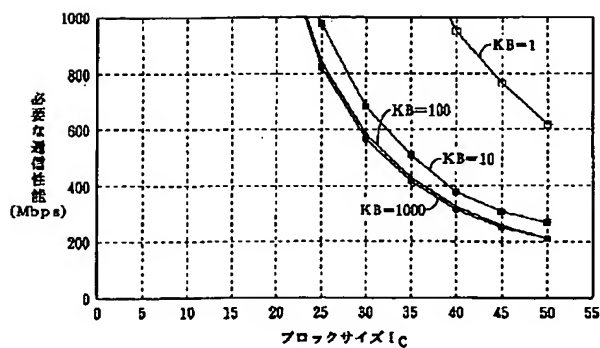
【図8】



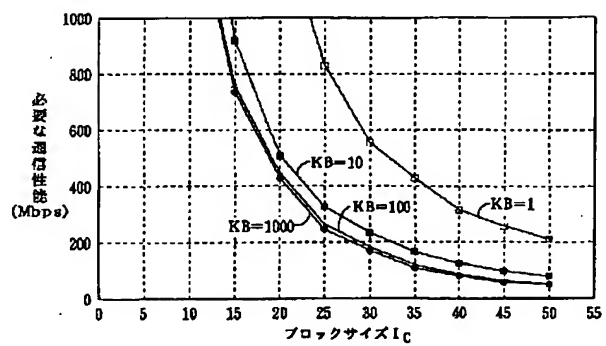
【図10】



【図11】



【図12】



【図 13】

(数式 1)

$$\phi_{\mu} = \sum_1 x_1 c_{1\mu}$$

(数式 2)

$$\Psi(1, 2, \dots, 2n) = \frac{1}{\sqrt{(2n)!}} \times$$

$$\begin{vmatrix} \phi_1^{\alpha(1)} & \phi_1^{\beta(1)} & \phi_2^{\alpha(1)} & \phi_2^{\beta(1)} & \cdots & \phi_n^{\alpha(1)} & \phi_n^{\beta(1)} \\ \phi_1^{\alpha(2)} & \phi_1^{\beta(2)} & \phi_2^{\alpha(2)} & \phi_2^{\beta(2)} & \cdots & \phi_n^{\alpha(2)} & \phi_n^{\beta(2)} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_1^{\alpha(2n)} & \phi_1^{\beta(2n)} & \phi_2^{\alpha(2n)} & \phi_2^{\beta(2n)} & \cdots & \phi_n^{\alpha(2n)} & \phi_n^{\beta(2n)} \end{vmatrix}$$

(数式 3)

$$H = H_1 + H_2$$

(数式 4)

$$H_1 = \sum_p \left[-\frac{1}{2} \nabla_p^2 - \sum_A \frac{Z_A}{r_{pA}} \right]$$

(数式 5)

$$H_2 = \sum_p \sum_{q \in p} \frac{1}{r_{pq}}$$

【図 14】

(数式 6)

$$\begin{aligned} \varepsilon &= \int \Psi H \Psi d\tau = \int \Psi (H_1 + H_2) \Psi d\tau \\ &= 2 \sum_{\mu} H_{\mu} + \sum_{\mu} \sum_{\nu} (2 J_{\mu\nu} - K_{\mu\nu}) \end{aligned}$$

(数式 7)

$$H_{\mu} = \int \phi_{\mu}(1) h^{\text{core}}(1) \phi_{\mu}(1) d\tau_1$$

(数式 8)

$$J_{\mu\nu} = \iint \phi_{\mu}(1) \phi_{\nu}(2) \frac{1}{r_{12}} \phi_{\mu}(1) \phi_{\nu}(2) d\tau_1 d\tau_2$$

(数式 9)

$$K_{\mu\nu} = \iint \phi_{\mu}(1) \phi_{\nu}(2) \frac{1}{r_{12}} \phi_{\mu}(2) \phi_{\nu}(1) d\tau_1 d\tau_2$$

【図15】

(数式10)

$$\varepsilon = 2 \sum_{\mu} \left[\sum_I \sum_J C_{I\mu} C_{J\mu} H_{IJ} \right] \\ + \sum_{\mu} \sum_{\nu} \left\{ \sum_I \sum_J \sum_K \sum_L C_{I\mu} C_{J\mu} C_{K\nu} C_{L\nu} [2G(I, J, K, L) - G(I, K, J, L)] \right\}$$

(数式11)

$$H_{IJ} = \int \chi_I(1) h^{\text{core}}(1) \chi_J(1) d\tau_1$$

(数式12)

$$h^{\text{core}}(1) = -\frac{1}{2} \nabla_1^2 - \sum_A \frac{Z_A}{r_{1A}}$$

(数式13)

$$G(I, J, K, L) = \iint \chi_I(1) \chi_J(1) \frac{1}{r_{12}} \chi_K(2) \chi_L(2) d\tau_1 d\tau_2$$

【図16】

(数式14)

$$\sum_J (F_{IJ} - \varepsilon_{\mu} S_{IJ}) C_{J\mu} = 0$$

(数式15)

$$F_{IJ} = H_{IJ} + \sum_K \sum_L P_{KL} \left[G(I, J, K, L) - \frac{1}{2} G(I, K, J, L) \right]$$

(数式16)

$$\varepsilon_{\mu} = H_{\mu} + \sum_{\nu} (2J_{\mu\nu} - K_{\mu\nu})$$

(数式17)

$$S_{IJ} = \int \chi_I(1) \chi_J(1) d\tau_1$$

(数式18)

$$P_{KL} = 2 \sum_{\nu} C_{K\nu} C_{L\nu}$$

(数式19)

$$G(I, J, K, L) = G(I, J, L, K) = G(J, I, K, L) = G(J, I, L, K) = \\ G(K, L, I, J) = G(L, K, I, J) = G(K, L, J, I) = G(L, K, J, I)$$

【図 17】

(数式 20)

$$\chi(r, n, R) = (r_x - R_x)^{n_x} (r_y - R_y)^{n_y} (r_z - R_z)^{n_z} \\ \times \sum_m d_m \exp[-\zeta_m (r - R)^2]$$

(数式 21)

$$\phi(r, n, R) \\ = (r_x - R_x)^{n_x} (r_y - R_y)^{n_y} (r_z - R_z)^{n_z} \exp[-\zeta (r - R)^2]$$

(数式 22)

$$\sum_m d_m \exp[-\zeta_m (r - R)^2]$$

(数式 23)

$$G(I, J, K, L) = \sum_{m1} \sum_{m2} \sum_{m3} \sum_{m4} d_{m1} d_{m2} d_{m3} d_{m4} g(i, j, k, l)$$

(数式 24)

$$g(i, j, k, l) = \int \int \phi_i(1) \phi_j(1) \frac{1}{r_{12}} \phi_k(2) \phi_l(2) d\tau_1 d\tau_2$$

(数式 25)

$$g(i, j, k, l) = g(i, j, l, k) = g(j, i, k, l) = g(j, i, l, k) = \\ g(k, l, i, j) = g(l, k, i, j) = g(k, l, j, i) = g(l, k, j, i)$$

【図 18】

(数式 26)

$$[0_a 0_b, 0_c 0_d]^{(m)} \\ = \frac{1}{\sqrt{\zeta + \eta}} \cdot K(\zeta_a, \zeta_b, A, B) \cdot K(\zeta_c, \zeta_d, C, D) \cdot F_m(T)$$

(数式 27)

$$F_m(T) = \int t^{2m} \exp(-T t^2) dt$$

(数式 28)

$$T = \rho (P - Q)^2$$

(数式 29)

$$K(\zeta, \zeta', R, R') \\ = \sqrt{2} \frac{\pi^{5/4}}{\zeta + \zeta'} \exp\left[-\frac{\zeta \zeta' (R - R')^2}{\zeta + \zeta'}\right]$$

(数式 30)

$$\zeta = \zeta_a + \zeta_b$$

(数式 31)

$$\eta = \zeta_c + \zeta_d$$

(数式 32)

$$\rho = \frac{\zeta \eta}{\zeta + \eta}$$

(数式 33)

$$P = \frac{\zeta_a A + \zeta_b B}{\zeta_a + \zeta_b}$$

(数式 34)

$$Q = \frac{\zeta_c C + \zeta_d D}{\zeta_c + \zeta_d}$$

【図 19】

(数式 35)

$$\begin{aligned}
 & [(a + l_i) b, c d]^{(m)} \\
 &= (P_i - A_i) [a b, c d]^{(m)} + (W_i - P_i) [a b, c d]^{(m+1)} \\
 &+ \frac{N_i(a)}{2\zeta} \left\{ [(a - l_i) b, c d]^{(m)} - \frac{\rho}{\zeta} [(a - l_i) b, c d]^{(m+1)} \right\} \\
 &+ \frac{N_i(b)}{2\zeta} \left\{ [a (b - l_i), c d]^{(m)} - \frac{\rho}{\zeta} [a (b - l_i), c d]^{(m+1)} \right\} \\
 &+ \frac{N_i(c)}{2(\zeta + \eta)} [a b, (c - l_i) d]^{(m+1)} \\
 &+ \frac{N_i(d)}{2(\zeta + \eta)} [a b, c (d - l_i)]^{(m+1)}
 \end{aligned}$$

(数式 36)

$$W = \frac{\zeta P + \eta Q}{\zeta + \eta}$$

(数式 37)

$$\exp = \left[-\frac{\zeta \zeta' (R - R')^2}{\zeta + \zeta'} \right]$$

【図 20】

(数式 38)

$$I \geq J, K \geq L, (IJ) \geq (KL)$$

$$(IJ) = \frac{I(I-1)}{2} + J, (KL) = \frac{K(K-1)}{2} + L$$

(数式 39)

$$F_{JJ} = -P_{KL} G(I, J, K, L),$$

$$F_{IK} = -\frac{1}{2} P_{JL} G(I, J, K, L),$$

$$F_{IL} = -\frac{1}{2} P_{JK} G(I, J, K, L),$$

$$F_{JK} = -\frac{1}{2} P_{IL} G(I, J, K, L),$$

$$F_{JL} = -\frac{1}{2} P_{IK} G(I, J, K, L),$$

$$F_{IL} = P_{IJ} G(I, J, K, L)$$

【図 21】

(数式 40)

$$I \geq J \geq K \geq L$$

(数式 41)

$$\begin{aligned} F_{IJ} &= P_{IL} [G(I, J, K, L) + G(I, K, J, L) + G(I, L, J, K)], \\ F_{IK} &= P_{JL} [G(I, J, K, L) + G(I, K, J, L) + G(I, L, J, K)] / 2, \\ F_{IL} &= P_{JK} [G(I, J, K, L) + G(I, K, J, L) + G(I, L, J, K)] / 2, \\ F_{JK} &= P_{IL} [G(I, J, K, L) + G(I, K, J, L) + G(I, L, J, K)] / 2, \\ F_{JL} &= P_{IK} [G(I, J, K, L) + G(I, K, J, L) + G(I, L, J, K)] / 2, \\ F_{KL} &= P_{IJ} [G(I, J, K, L) + G(I, K, J, L) + G(I, L, J, K)] \end{aligned}$$

(数式 42)

$$I_B \geq J_B \geq K_B \geq L_B$$

(数式 43)

$$I_B \geq J_B, K_B \geq L_B, (I_B J_B) \geq (K_B L_B)$$

【図 22】

(数式 44)

$$\frac{2(2I_C^2 + K_B I_C^2) \times 64 \times M}{T_{eri} \times \alpha^2 K_B I_C^4} = \frac{(2 + K_B) \times 128 M}{T_{eri} \alpha^2 K_B I_C^2} \quad (\text{Mbp s})$$

(数式 45)

$$\frac{2(2I_C^2 + K_B I_C^2) \times 64 \times M}{T_{eri} \times 3 \alpha^2 K_B I_C^4} = \frac{(2 + K_B) \times 128 M}{3 T_{eri} \alpha^2 K_B I_C^2} \quad (\text{Mbp s})$$

(数式 46)

$$(I_C \times I_D) [1 - (1 - \alpha)^{I_C}] \quad (\text{個})$$

(数式 47)

$$(I_C \times I_D) [1 - (1 - \alpha)] \quad (\text{個})$$

【図 23】

(数式 48)

$$2 \times (I_C \times I_D) [3 - (1 - \alpha)^{I_C} - (1 - \alpha)^{I_C} - (1 - \alpha)] \times 64 \quad (\text{bit})$$

(数式 49)

$$\frac{2 \times (I_C \times I_D) [3 - (1 - \alpha)^{I_C} - (1 - \alpha)^{I_C} - (1 - \alpha)] \times 64 \times M}{T_{\text{eri}} \times \alpha^2 I_C^3 I_D} \quad (\text{Mbps})$$

(数式 50)

$$(I_C \times I_D) [1 - (1 - \alpha)^{2I_C + 1}] \quad (\text{個})$$

(数式 51)

$$2 \times 3 (I_C \times I_D) [1 - (1 - \alpha)^{2I_C + 1}] \times 64 \quad (\text{bit})$$

(数式 52)

$$\frac{2 \times (I_C \times I_D) [1 - (1 - \alpha)^{2I_C + 1}] \times 64 \times M}{T_{\text{eri}} \times 3 \alpha^2 I_C^3 I_D} \quad (\text{Mbps})$$

フロントページの続き

- (72)発明者 稲畑 深二郎
神奈川県足柄上郡中井町境430 グリーン
テクなかい富士ゼロックス株式会社内
- (72)発明者 宮川 宣明
神奈川県足柄上郡中井町境430 グリーン
テクなかい富士ゼロックス株式会社内
- (72)発明者 高島 一
東京都豊島区高田 3-24-1 大正製薬株
式会社内
- (72)発明者 北村 一泰
東京都豊島区高田 3-24-1 大正製薬株
式会社内

Fターム(参考) 5B045 AA07 BB12 BB47 GG12
5B056 AA04 BB31 BB72 HH00